



Titre: Synthèse de réseaux pour des communications multipoint
Title:

Auteur: Aref Meddeb
Author:

Date: 1998

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Meddeb, A. (1998). Synthèse de réseaux pour des communications multipoint
Citation: [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie.
<https://publications.polymtl.ca/6795/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/6795/>
PolyPublie URL:

**Directeurs de
recherche:**
Advisors:

Programme: Non spécifié
Program:

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI[®]

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UNIVERSITÉ DE MONTRÉAL

SYNTHÈSE DE RÉSEAUX POUR DES COMMUNICATIONS MULTIPONT

AREF MEDDEB

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR (Ph.D.)
(GÉNIE ÉLECTRIQUE)

décembre 1998



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-38726-7

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée:

SYNTHÈSE DE RÉSEAUX POUR DES COMMUNICATIONS MULTIPOINT

présentée par: MEDDEB Aref

en vue de l'obtention du diplôme de: Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de:

M. HACCOUN David, Ph.D., Président

Mme ROSENBERG Catherine, Doc.Sc., directrice de recherche

M. GIRARD André, Ph.D., codirecteur de recherche

Mme SANSÓ Brunilde, Ph.D., membre

M. DZIONG Zbigniew, Ph.D., membre

À ma mère Aïcha, mon père Abdelaziz, mon frère Adnane et ma petite sœur Afef.

REMERCIEMENTS

Je tiens en premier lieu à remercier au même titre mes codirecteurs de thèse, les Dr. Catherine Rosenberg et Dr. André Girard. La collaboration fructueuse de Catherine et André a constitué l'une des clés de réussite de ce travail de recherche. Malgré leurs emplois du temps très chargés, j'ai dû occuper pour de longs moments une place privilégiée dans leurs agendas et je leur en suis reconnaissant.

Je tiens à remercier Catherine pour avoir bien voulu accepter de diriger mon doctorat à l'automne 1995, et de m'avoir proposé un sujet de recherche d'actualité et aussi intéressant. Le renouvellement de confiance de Catherine en moi (après la maîtrise) a contribué considérablement à ma motivation pour entamer un projet de doctorat. La compétence de Catherine, son excellent jugement et la qualité exceptionnelle de son encadrement m'ont beaucoup aidé à amener à terme mes travaux de recherche. Je tiens aussi à remercier Catherine de m'avoir octroyé une aide financière et d'avoir mis à notre disposition (au laboratoire de réseaux multi-services à large bande) du matériel informatique très performant, qui étaient ô combien indispensables pour le bon déroulement de mes travaux de recherche.

Je tiens également à remercier André, en particulier, pour sa compétence, son savoir faire, la qualité exceptionnelle de son encadrement et son expertise dans le domaine de conception de réseaux de télécommunications qu'il a mis à mon service

tout au long de ces trois dernières années. Les nombreuses rencontres de travail que nous avons eues ont été très bénéfiques pour moi. Je lui dois une grande partie de ce que je sais aujourd'hui et je lui suis particulièrement reconnaissant pour m'avoir suggéré le modèle de synthèse de réseaux avec communications multipoint, qui a constitué le point de départ de mon travail de recherche.

Je voulais aussi remercier André et Catherine pour m'avoir donné l'occasion de participer à de nombreuses conférences et séminaires qui ont été bénéfiques pour ma formation.

Je tiens aussi à remercier les professeurs qui ont bien voulu accepter de constituer mon jury et de l'intérêt et de l'encouragement qu'ils ont manifestés tout au long de mon doctorat, les Dr. David Haccoun, Dr. Brunilde Sansó et Dr. Gilles Deslauriers. J'aimerais aussi remercier le Dr. Zbigniew Dziong qui a bien voulu jouer le rôle de l'examineur externe et le Dr. Maurice Amram, qui a bien voulu représenter le doyen de la Faculté des Études Supérieures de l'université de Montréal.

J'aimerais remercier les étudiants et chercheurs du laboratoire de réseaux multi-services à large bande qui m'ont accompagné pendant les années que j'ai passées à Poly et avec qui j'ai passé d'agréables moments. En particulier, je tiens à remercier Verna Friesen avec qui j'ai travaillé durant la première année de mon doctorat, Charles Levert qui a été d'une aide précieuse quant à l'administration de notre réseau local, Josée Mignault, Haïkel Yaïche, Abdelkrim Mekkaoui, Valery Augais et Sylvain Delas pour les quelques discussions fructueuses que j'ai eues avec eux.

J'aimerais aussi remercier tous les professeurs et personnel du département de

génie électrique et de génie informatique qui m'ont aidé tout au long de mon doctorat. Merci en particulier aux professeurs Louis Granger et Michel Lemire ainsi qu'à Martine Pinsonneault et à Nathalie Lévesque.

Je tiens aussi à remercier tous les amis que j'ai connus à Montréal ces cinq dernières années ainsi que les autres. En particulier, Louay avec qui j'ai partagé mon logement pendant les quatre dernières années à Montréal et de la part de qui j'ai beaucoup appris.

Enfin, j'aimerais remercier de tout mon coeur, tous les membres de ma famille qui ont été d'un support moral exceptionnel. Sans leur aide, il aurait été difficile de se concentrer sur mon travail de recherche. Merci Maman, Papa, Adnane et Afef d'avoir toujours été là et d'avoir contribué considérablement à mon succès. Notamment, merci Papa et Maman pour les petits gâteaux que vous m'envoyez de temps en temps de Tunis et qui m'ont procuré de l'énergie, ô combien nécessaire pour les longues journées de travail.

RÉSUMÉ

Ce travail de recherche porte sur les réseaux offrant des communications multipoint. Nous nous intéressons essentiellement à leur synthèse, ce qui comprend deux sous-problèmes, l'acheminement des appels et le dimensionnement des artères de transmission. L'acheminement consiste à choisir les routes sur lesquelles les appels seront connectés dans le but de maximiser le revenu généré par les connexions, dans un réseau donné et pour des demandes de trafic données. Le dimensionnement consiste à optimiser les capacités des artères de transmission dans le but de minimiser le coût du réseau, pour des demandes de trafic données. Le problème de synthèse consiste alors à maximiser la valeur nette du réseau soit, la différence entre le revenu généré par les connexions et le coût du réseau.

Le problème de synthèse de réseaux a fait l'objet de nombreuses études dans le contexte de réseaux n'offrant que des communications point-à-point (commutation de circuit) impliquant une paire d'utilisateurs chacune. Ces études ont montré que connecter les appels d'abord sur l'arc direct est généralement optimal, non seulement en terme de revenu d'appels, dans le contexte d'acheminement, mais aussi en terme de valeur nette du réseau, dans le contexte de synthèse de réseau. Dans le contexte de réseaux offrant des communications multipoint, on ne parle plus de paire d'utilisateurs voulant se connecter mais plutôt d'un ensemble d'utilisateurs. On utilise

alors des arbres plutôt que des chemins pour connecter les appels et on ne sait pas comment sélectionner ces arbres (c.-à-d., former les ensembles d'arbres candidats sur lesquels les appels seront connectés) de façon à maximiser le revenu ou la valeur nette (différence entre le revenu et le coût) du réseau.

Notre premier objectif consiste à comprendre, dans un contexte mono-débit, le problème de sélection d'arbres, d'évaluer son impact sur les performances du réseau et éventuellement, trouver des critères simples pour cette sélection d'arbres. Le second objectif consiste à proposer une formulation mathématique pour le problème de sélection d'arbres et à proposer des méthodes numériques pour la résolution de ce problème dans les contextes d'acheminement et de synthèse de réseaux. Le troisième objectif consiste à étudier la sensibilité du réseau à la variation du trafic (variation des demandes de trafic ou des ensembles de participants aux communications), en termes de perte de revenu et de probabilité de perte d'appels. Nous voulons, entre autre, évaluer l'importance de la connaissance à priori du trafic car en pratique, il est très difficile de le prévoir. Enfin, le dernier objectif consiste à généraliser l'étude au cas de réseaux multi-débit. En utilisant la notion de débit équivalent, nous pouvons alors étudier le problème de synthèse de réseaux à large bande de type ATM avec communications multipoint.

Nous pouvons résumer les contributions réalisées dans le cadre de ce travail de recherche comme suit. Nous avons tout d'abord formulé le problème de synthèse de réseaux offrant des communications multipoint et avons dégagé les points essentiels à étudier. Nous avons proposé un algorithme pour résoudre le problème d'acheminement et un algorithme pour résoudre le problème de dimensionnement, pour un ensemble d'arbres donné. Nous avons utilisé à cette fin des méthodes d'optimisation

classiques notamment, la méthode du gradient projeté pour l'optimisation de l'acheminement et la relaxation Lagrangienne pour l'optimisation du dimensionnement.

Nous avons ensuite établi un critère simple de sélection d'arbres pour l'acheminement d'une part, et un autre pour la synthèse de réseaux, d'autre part. Nous avons montré que ces critères suggéraient des actions contraires puisque selon l'un, il faut disperser le trafic dans le contexte d'acheminement c.-à-d., utiliser des arbres disjoints alors que selon l'autre, il faut concentrer le trafic dans le contexte de synthèse c.-à-d., utiliser des arbres ayant plusieurs arcs en commun. Nous avons alors expliqué les raisons et les conséquences d'un tel dilemme.

Nous avons ensuite proposé deux algorithmes de sélection d'arbres dans le contexte d'acheminement d'appel dont un, basé sur la méthode du gradient projeté avec génération de colonnes, qui donne une solution exacte (optimum local) et un autre algorithme plus simple, basé sur une heuristique. Nous avons montré que l'algorithme qui donne une solution exacte est très complexe et qu'il ne peut être utilisé pour des réseaux de taille raisonnable alors que celui basé sur l'heuristique donne rapidement des solutions comparables à celles obtenues par la méthode exacte. À l'aide de ces algorithmes, nous avons illustré que l'utilisation de plusieurs arbres disjoints de petite taille pour connecter les appels est souvent optimale.

Nous avons également proposé deux algorithmes de sélection d'arbres dans le contexte de synthèse de réseaux dont un est basé sur la méthode du gradient projeté avec génération de colonnes et l'autre sur une heuristique simple. Nous avons montré que le premier algorithme est inutilisable en pratique en raison de sa complexité alors que le second conduit à des solutions quasi optimales. À l'aide de cet algorithme, nous avons montré que pour des appels multipoint, les arbres optimaux (en terme de

synthèse) utilisés pour connecter les appels multipoint ont plusieurs arcs en commun faisant en sorte que le réseau optimal est très faiblement connecté, se réduisant souvent à un arbre.

Nous avons ensuite étudié le problème de synthèse de réseaux offrant des communications point-à-point et point-à-multipoint et nous avons montré que les résultats dépendaient de l'intensité relative des trafics: plus le trafic point-à-point est intense par rapport au trafic point-à-multipoint, plus le réseau optimal est fortement connecté. Dans le cas contraire, où le trafic point-à-multipoint est plus intense, on a tendance à obtenir un réseau faiblement connecté. En général, la connectivité du réseau varie entre celle d'un arbre et celle d'un graphe complet.

Nous avons aussi étudié le problème de sensibilité de la performance du réseau obtenu par synthèse optimale à la variation de trafic. Nous avons montré qu'en général, la variation de l'intensité des demandes de trafic n'affecte que très légèrement la performance du réseau alors que la variations des membres participant aux communications peut en affecter considérablement la performance. Nous avons alors proposé des moyens pour réagir aux variations de trafic notamment, ré-optimiser l'acheminement des appels, re-dimensionner les artères de transmission et comme dernier recours, refaire la synthèse du réseau avec les nouvelles matrices de trafic.

D'autre part, dans la mesure où pour des appels multipoint le réseau optimal (obtenu après synthèse) est très faiblement connecté, en cas de variation de trafic, la ré-optimisation de l'acheminement est très difficile, voire, impossible. Nous avons alors proposé un algorithme qui permet d'augmenter la connectivité du réseau de façon économique et nous avons montré que l'augmentation de la connectivité du

réseau est beaucoup plus coûteuse que l'augmentation du revenu générée en ré-optimisant l'acheminement.

Enfin, nous avons généralisé notre étude au cas de réseaux multi-débit et nous avons montré que nous obtenons pratiquement les mêmes conclusions notamment, il faut disperser le trafic dans le contexte d'acheminement et le concentrer dans le contexte de synthèse de réseaux. En utilisant le concept de débit équivalent, nous pouvons alors utiliser nos résultats dans le contexte de synthèse de réseaux à large bande de type ATM.

ABSTRACT

This work deals with the design of networks supporting multipoint communications. The network design problem includes two subproblems, routing and dimensioning. The routing is to choose the routes that carry the calls in order to maximize the revenue generated by the multicast connections for a given network and set of traffic demands. The dimensioning is to optimize the capacities of the network links in order to minimize the cost of the network for a given set of traffic demands. The design problem is to maximize the net value of the network which is the difference between the revenue and the cost of the network.

Network design with point-to-point connections has been studied extensively in the literature and it was shown that the direct link is often optimal to connect the calls either in the call routing context or in the dimensioning context.

Multipoint connections are connected using trees instead of paths and we do not know how to select the trees (i.e., build the candidate sets of trees) that should be used to connect the calls in order to maximize the revenue or the net value of the network.

Our first objective is to understand the tree selection problem and to evaluate its impact on the network performance and eventually, find simple criteria upon which the tree selection should be based. Second, we want to give a mathematical

formulation for the tree selection problem and develop numerical algorithms to solve it. Third, we want to investigate how traffic variation (either of the traffic demands or of the members participating in the communications) affects the network performance and how should we react to such variation. We also want to investigate the importance of knowing a priori the traffic demands and the members participating in the communications since in practice, the traffic is difficult to forecast. Finally, we want to generalize our work to the multi-class case which, through the equivalent bandwidth concept, would allow us to apply our work to ATM-like broadband networks.

We can summarize our contributions as follows. First, we have highlighted and prioritized the important points of the network design problem with multipoint communications. We have proposed two algorithms to optimize the call routing and the network design for a given set of trees. We have used classical optimization methods i.e., the gradient projection method for the call routing problem and the Lagrangian relaxation method for the network design problem.

We have established two simple criteria for the tree selection, one for routing and one for dimensioning. We have shown that these two criteria are contradicting each other i.e., we should use disjoint trees for routing (split the traffic) and trees with many links in common for dimensioning (concentrate the traffic) and we have explained why we have such a dilemma and discussed its consequences.

We then have proposed two algorithms to select the trees in the call routing context, one based on the gradient projection method with column generation which yields an exact solution (local optimum) and one based on a heuristic. We have shown that the first algorithm is very complex and that it cannot be used in practice, for

any network of a reasonable size, while the second algorithm can be used in practice to quickly select trees and has a performance close to the exact one. By means of these algorithms, we have shown that for the call routing to be optimal, we should use as many small and disjoint trees as possible to connect the calls.

We have also proposed two algorithms to select the trees in the network design context, one based on the gradient projection method with column generation and one based on a simple heuristic. We have shown that the first algorithm is very complex and that it cannot be used in practice and that the second algorithm yields nearly optimal tree selections. By means of this algorithm, we have illustrated that in the network design context, the optimal trees share many links resulting in a weakly connected network that is often reduced to a tree.

We have also studied the network design problem with point-to-point and point-to-multipoint communications and have shown that the results depend on the mix of traffic: the optimal network topology can have any connectivity between that of a tree and that of a fully connected network, depending on the intensity of both types of traffic. The more intense the point-to-point traffic, the higher the network connectivity.

We have also studied the sensitivity of the network performance in the presence of traffic variation. We have shown that in general, the variation of the traffic demands intensity has a negligible impact on the network performance whereas the variation of the members participating in the multipoint communications has a significant impact on the network performance. We then have proposed different ways to react to the traffic variation i.e., re-optimize the call routing, update the link capacities (dimensioning) or less frequently, re-design the network for the new traffic pattern.

In addition, since when we have multipoint communications, the optimal network that is obtained by solving the design problem is weakly connected, it is very difficult and even impossible to re-optimize the call routing when the traffic varies. We then have proposed an algorithm to economically increase the network connectivity and we have shown that introducing more connectivity in the network is done at a much higher price than the improvement of the revenue obtained by re-optimizing the call routing.

Finally, we have generalized our study to the multi-class case and we have shown that our conclusions are valid in that context, i.e., splitting the traffic in the call routing context and concentrating it in the design context are optimal. This allows us to apply our work to ATM like broadband networks through the equivalent bandwidth concept.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	viii
ABSTRACT	xiii
TABLE DES MATIÈRES	xvii
LISTE DES TABLEAUX	xxiii
LISTE DES FIGURES	xxvi
LISTE DES ANNEXES	xxix
LISTE DES ABRÉVIATIONS	xxx
LISTE DES SYMBOLES	xxxiii
CHAPITRE 1 : Introduction	1
1.1 Problématique	5
1.1.1 Le problème de sélection d'arbres	7

1.1.2	Point-à-point vs point-à-multipoint	8
1.1.3	Sensibilité à la variation du trafic	9
1.1.4	Multi-débit	10
1.2	Objectifs et contributions	10
1.2.1	Objectifs	11
1.2.2	Contributions	12
1.3	Organisation de la thèse	13
CHAPITRE 2 : Revue de la littérature		14
2.1	Le multipoint	15
2.1.1	Commutateur duplicateur	15
2.1.2	Aspect dynamique des connexions	17
2.1.3	Qualité de service	19
2.1.4	Adressage multipoint	20
2.1.5	Tarification	21
2.2	Synthèse de réseaux	22
2.2.1	Demandes de trafic	25
2.2.2	Algorithmes de génération d'arbres	29
2.2.3	Techniques d'acheminement d'appels	34
2.2.4	Acheminement de paquets	38
2.3	Le multipoint dans Internet	39
2.3.1	Le réseau Mbone	39
2.3.2	Protocole UDP	41
2.3.3	Protocole IGMP	41
2.3.4	Acheminement	42

2.3.5	Qualité de service	44
2.3.6	Adressage	45
2.3.7	Identification des membres	45
2.3.8	Résumé	46
2.4	Le multipoint dans l'ATM	46
2.4.1	Définitions de l'UIT-T	47
2.4.2	Définitions de l'ATM Forum	49
2.4.3	Entrelacement de cellules	50
2.4.4	Décomposition de connexions multipoint	53
2.4.5	Qualité de service	55
2.4.6	Adressage	56
2.4.7	Résumé	57
2.5	Trafic Internet multipoint sur ATM	57
2.5.1	Adressage	58
2.5.2	Comment réaliser le multipoint-à-multipoint?	59
2.5.3	Résumé	62
2.6	Conclusion	63
CHAPITRE 3 : Modélisation du problème de synthèse de réseaux		64
3.1	Problèmes multipoint pertinents	65
3.1.1	Classification des communications multipoint	65
3.1.2	Point-à-multipoint	66
3.1.3	Multipoint-à-multipoint	67
3.1.4	Multipoint-à-point	69
3.1.5	Temps réel	70

3.1.6	Résumé	71
3.2	Description du modèle	71
3.2.1	Acheminement	73
3.2.2	Calcul du blocage	74
3.2.3	Optimisation de l'acheminement	77
3.2.4	Calcul des multiplicateurs	80
3.3	Synthèse de réseaux	81
3.4	Algorithmes d'optimisation pour un ensemble d'arbres donné	83
3.4.1	Algorithme d'optimisation du partage de charge	84
3.4.2	Algorithme de dimensionnement	89
3.5	Conclusion	90
CHAPITRE 4 : Sélection d'arbres pour l'acheminement		92
4.1	Formulation du problème de sélection d'arbres	93
4.2	Impact de la sélection d'arbres sur l'acheminement	95
4.2.1	Définitions	95
4.2.2	Résumé	111
4.3	Algorithmes de sélection d'arbres	114
4.3.1	Méthode exacte	114
4.3.2	Méthode heuristique	119
4.3.3	Résultats pour d'autres réseaux	132
4.3.4	Résumé	137
4.4	Acheminement en temps réel	138
4.5	Autres méthodes d'optimisation de la sélection d'arbres	140
4.5.1	Algorithmes génétiques	140

4.5.2	Recherche de type tabou	142
4.5.3	Résumé	143
4.6	Conclusion	143
CHAPITRE 5 : Sélection d'arbres pour la synthèse de réseaux . . .		145
5.1	Formulation du problème	146
5.2	Critère de sélection d'arbres	147
5.2.1	Influence du facteur de coût C_0 et des revenus $w^k(c)$	153
5.2.2	Interprétation des résultats	156
5.2.3	Résumé	157
5.3	Méthode exacte	158
5.3.1	Gradient projeté	158
5.3.2	Génération de colonnes	159
5.4	Méthode heuristique	161
5.4.1	Approximation du coût et des contraintes de QS	161
5.4.2	Concentration de trafic	162
5.4.3	Algorithme de sélection d'arbres	167
5.4.4	Algorithme de concentration globale de trafic	168
5.4.5	Concentration individuelle du trafic	172
5.4.6	Dilemme point-à-point/multipoint	177
5.4.7	Résultats pour d'autres réseaux	182
5.5	Variation du trafic	194
5.6	Dilemme acheminement/dimensionnement	202
5.7	Utilité pratique de l'heuristique de sélection d'arbres	209
5.8	Autres méthodes d'optimisation de la sélection d'arbres	210

5.8.1	Algorithmes génétiques	210
5.8.2	Recherche de type tabou	211
5.8.3	Résumé	212
5.9	Conclusion	212
CHAPITRE 6 : Conclusions		215
6.1	Résumé des contributions	215
6.1.1	Sélection d'arbres	216
6.1.2	Variation du trafic	217
6.1.3	Dilemme point-à-point/multipoint	218
6.1.4	Dilemme dimensionnement/acheminement	219
6.1.5	Multi-débit	220
6.2	Autres problèmes pertinents	220
RÉFÉRENCES		226
ANNEXES		238

LISTE DES TABLEAUX

4.1	Résultats du partage de charge pour un seul type d'appel.	99
4.2	Variables d'arcs avec un seul type d'appel.	99
4.3	Résultats du partage de charge pour le nouveau type d'appel.	101
4.4	Résultats pour les arcs avec le nouveau type d'appel.	102
4.5	Données.	104
4.6	Acheminement optimal générant un revenu total de 68.3.	106
4.7	Variables d'arcs optimales pour les arcs coûteux ($ r_s \geq 0.14$).	106
4.8	Données pour le multi-débit.	111
4.9	Variables d'arcs optimales pour les arcs coûteux pour le multi-débit ($ r_s(c) \geq 0.15$).	113
4.10	Résultats pour différentes valeurs de n et \overline{M}	128
4.11	Acheminement des appels point-à-multipoint.	130
4.12	QS pour le multi-débit.	132
4.13	Acheminement des appels point-à-multipoint avec du multi-débit. . .	132
4.14	Résultats pour différents nombre de nœuds pour $K = 10$	133
4.15	Résultats pour différents nombre de types d'appels pour $N = 16$. . .	134

4.16 Résultats pour différentes classes d'appel pour $N = 9$ et 10 types d'appel.	135
4.17 $\max_k L^k(c)$ pour des réseau multi-débit.	136
5.1 QS pour le dimensionnement optimal du réseau.	148
5.2 Influence des arcs communs sur la synthèse de réseaux.	151
5.3 QS pour le dimensionnement optimal du réseau avec du multi-débit.	152
5.4 Acheminement et QS pour le dimensionnement du réseau vs C_0	154
5.5 Capacités (N_s) des arcs et coût du réseau vs C_0	155
5.6 Synthèse de réseaux avec du point-à-multipoint.	171
5.7 Variables d'arcs.	172
5.8 Synthèse de réseaux avec du multi-débit.	173
5.9 QS pour la synthèse du réseau avec du multi-débit.	173
5.10 Variables d'arcs pour le multi-débit (classe 2 seulement).	174
5.11 Capacités des arcs avec un appel point-à-point et trois appels point- à-multipoint.	178
5.12 Types d'appel point-à-point.	179
5.13 Synthèse de réseaux avec point-à-point et point-à-multipoint.	181
5.14 (η_g, η_i, η) pour des réseaux multi-débit.	186
5.15 $\max_k L^k(c)$ obtenues par concentration du trafic pour des réseaux multi-débit.	186
5.16 Variables d'arcs pour la classe 3 avec trois classes d'appel et $\rho = 2/3$	187
5.17 Comparaison de η_g et η'_g	190

5.18	Demandes de trafic point-à-multipoint et destinations prévues et réelles. Les destinations prévues et qui ne le devraient pas sont en petit caractère et les destinations réelles non prévues sont en caractères gras.	199
5.19	Demandes de trafic point-à-point prévues et réelles.	199
5.20	Performance du réseau en présence de variation de trafic avec exclusivement du trafic point-à-multipoint.	200
5.21	Performance du réseau en présence de variation de trafic avec du trafic point-à-multipoint et point-à-point. $k \leq 10$ correspond au appels point-à-multipoint.	201
5.22	Performance en présence de variation de trafic.	207
5.23	Nombre d'arcs dans le réseau vs N et ρ	208
5.24	Performance du réseau quand l'arc s_1 est remplacé par l'arc s_2	212

LISTE DES FIGURES

1.1	N appels point-à-point ($N = 4$ sur cette figure).	3
1.2	Appel point-à-multipoint vers N destinations avec des commutateurs duplicateurs ($N = 4$).	4
2.1	Exemple d'arbre de Steiner.	29
2.2	Exemple d'acheminement avec partage de charge.	35
2.3	Exemple d'acheminement avec débordement.	36
2.4	Topologie simplifiée de Mbone.	40
2.5	Types de communications multipoint.	47
2.6	Problème d'entrelacement de cellules.	51
2.7	Décomposition de connexions multipoint en ATM.	54
2.8	Solution avec serveurs duplicateurs.	60
2.9	Solution avec un ensemble de VC.	62
3.1	Le trafic dans le réseau commuté ne dépend pas du changement des membres.	67
3.2	Effet de la modification du groupe d'utilisateurs.	68
4.1	Topologie du réseau.	98

4.2	Choix d'arbres pour un seul type d'appel.	98
4.3	Choix d'arbres pour le nouveau type d'appel.	101
4.4	Topologie du réseau.	104
4.5	Arbres candidats et acheminement optimal (les valeurs en pourcent représentent les coefficients de partage de charge α_m^k). L'épaisseur d'un arc est proportionnelle au nombre d'arbres qui l'utilisent.	105
4.6	Sélection d'arbres et acheminement optimal avec un petit nombre d'arcs en commun (les valeurs en pourcent donnent le partage de charge optimal).	109
4.7	Arbres candidats pour le multi-débit. Pour chaque arbre, nous don- nons le coefficient de partage de charge en pourcent (coin gauche en bas) et le revenu net pondéré (coin droit en bas). Le revenu total est 131.9.	112
4.8	Sélection d'arbres avec la méthode heuristique.	131
5.1	Topologie optimale du réseau.	148
5.2	Sélection d'arbres ayant plusieurs arcs en commun.	150
5.3	Topologie optimale du réseau avec multi-débit.	152
5.4	Variation de la capacité en fonction du trafic offert à blocage constant ($B_s = 0.001$).	164
5.5	Concentration de trafic sur un seul chemin.	166
5.6	Topologie du réseau en arbre optimal.	171
5.7	Topologie du RAQO avec du multi-débit.	173
5.8	Utilisation de l'arc direct au lieu des arcs du RAQO.	175

5.9	Topologie optimale du réseau avec des appels point-à-point en utilisant les arbres directs.	179
5.10	Topologie optimale du réseau avec des appels point-à-point.	181
5.11	Variation de η_g (en %) vs N	183
5.12	Variation de η_i (en %) vs N	184
5.13	Variation de η (en %) vs N	184
5.14	η'_g en fonction de N	191
5.15	η'_g (en %) en fonction de N avec du point-à-point.	195
5.16	η'_i (en %) en fonction de N avec du point-à-point.	196
5.17	Augmentation de la connectivité du réseau.	206

LISTE DES ANNEXES

A	Preuve de la proposition de concentration de trafic	238
B	Preuve de la proposition du chemin unique	240

LISTE DES ABRÉVIATIONS

A

AAL: ATM Adaptation Layer

ABR: Available Bit Rate

ATM: Asynchronous Transfer Mode

C

CAC: Call Admission Control

CBR: Constant Bit Rate

CBT: Core Based Trees

D

DCR: Dynamically Controlled Routing

DNHR: Dynamic NonHierarchical Routing

F

FTP: File Transfer Protocol

I

IETF: Internet Engineering Task Force

IGMP: Internet Group Management Protocol

IP: Internet Protocol

M

MARS: Multicast Address Resolution Server

Mbone: Multicast backbone

MID: Multiplexing Identifier

MOSPF: MulticastOSPF

MST: Minimum Spanning Tree

N

RAQO: Réseau en Arbre Quasi Optimal

O

OOC: Originating Office Control

OSPF: Open Shortest Path First

P

PIM: Protocol Independent Multicast

Q

QS: Qualité de Service (niveau appel)

qs: qualité de service (niveau unité de données)

R

RNIS: Réseau Numérique à Integration de Services

RNIS-LB: RNIS à Large Bande

RSVP: Resource Reservation Protocol

S

SDP: Session Description Protocol

SEAM: Scalable and Efficient ATM Multipoint-to-multipoint

SMART: Shared MANY-to-many ReservaTion

SOC: Sequential Office Control

T

TCP: Transport Control Protocol

TOS: Type Of Service

U

UIT: Union Internationale des Télécommunications

UIT-T: UIT-secteur de standardisation des Télécommunications

V

VBR: Variable Bit Rate

VC: Virtual Channel

VCI: Virtual Channel Identification

VP: Virtual Path

VPI: Virtual Path Identification

W

WWW: World Wide Web

LISTE DES SYMBOLES

A

$A^k(c)$: Demande de trafic pour le type d'appel k de classe c

$\bar{A}^k(c)$: Trafic de type k et de classe c écoulé par le réseau

$a_s(c)$: Trafic de classe c offert à l'arc s

$\bar{a}_s(c)$: Trafic de classe c écoulé sur l'arc s

$\alpha_m^k(c)$: Coefficient de partage de charge pour l'arbre m de type k et de classe c

B

$B_s(c)$: Probabilité de blocage sur l'arc s pour la classe c

β : Pas de déplacement

C

C : Nombre de classes d'appel

C_0 : Constante donnant le coût de capacité linéaire

$C_s(N_s)$: Coût de capacité

c : Indice de classe d'appel

D

$D^k(c)$: Nombre de destinations pour le type k de classe c

d : Vecteur de direction de déplacement

E

η : Économie (en pourcent) par concentration de trafic

η_g : Économie (en pourcent) par concentration globale du trafic

η_i : Économie (en pourcent) par concentration individuelle du trafic

G

\mathcal{G} : Graphe du réseau

$g_s^c(\cdot)$: Fonction donnant le blocage sur l'arc s pour les appels de classe c

H

$h(\cdot)$: Fonction donnant le degré d'un nœud

h_{\min} : Degré minimal pour les nœuds du réseau

I

$\mathcal{I}_{s,(k,c,m)}$: Matrice d'incidence arc-arbre

J

$\mathcal{J}_{s,(k,c,m)}$: Matrice d'incidence arc-type

K

K : Nombre de types d'appel

k : Indice de type d'appel

L

$L^k(c)$: Probabilité de perte d'appel pour le type k de classe c

$L_m^k(c)$: Probabilité de perte d'appel sur l'arbre m pour le type k de classe c

$\overline{L}^k(c)$: Probabilité de perte d'appel maximale permise pour le type k de classe c

ℓ_s : Trafic maximal potentiel qui peut être offert à l'arc s

λ : Vecteur dual pour le gradient projeté

M

M : Nombre d'arbres

$M^k(c)$: Nombre d'arbres pour le type k et la classe c

M_W : Sous-espace de contraintes actives

$\overline{M}^k(c)$: Nombre d'arbres maximal pour le type k et la classe c

m : Indice d'arbre

N

N : Nombre de nœuds dans le réseau

N_s : Capacité (nombre de circuits) de l'arc s

n : Nombre de nœuds de Steiner

O

$O(\cdot)$: Fonction donnant la complexité de calcul

Ω : Ensemble de contraintes actives

P

P : Matrice de projection

P_l : Poids d'un arbre

$p_{k,c}$: Nombre de contraintes inactives pour le type k et la classe c

Q

$Q_m^k(c)$: Probabilité de connexion sur l'arbre m pour le type k de classe c

R

$R_m^k(c)$: Revenu marginal de l'arbre m pour le type k et la classe c

$\overline{R}_m^k(c)$: Revenu net pondéré de l'arbre m pour le type k et la classe c (acheminement)

$\overline{\overline{R}}_m^k(c)$: Revenu net de l'arbre m pour le type k et la classe c (synthèse)

$r_s(c)$: Coût induit de l'arc s pour la classe c

ρ : Rapport du nombre d'appels point-à-point au nombre d'appels point-à-multipoint

S

S : Ensemble contenant tous les arbres

$S^k(c)$: Ensemble d'arbres candidats pour le type k de classe c

s : Indice d'arc

T

$T^k(c)$: Ensemble pré-défini d'arbres candidats possibles pour le type k de classe c

U

$u_m^k(c)$: Multiplicateur de Lagrange pour la contrainte d'inégalité sur les $\alpha_m^k(c)$

V

$v^k(c)$: Multiplicateur de Lagrange pour la contrainte d'égalité sur les $\alpha_m^k(c)$

W

$W(c)$: Débit de la classe c

$w^k(c)$: Revenu des connexions de type k et de classe c

X

\hat{X} : Approximation donnant une borne supérieure sur une variable X

\check{X} : Approximation donnant une borne inférieure sur une variable X

\tilde{X} : Approximation d'une variable X

$x^k(c)$: Multiplicateur de Lagrange pour la contrainte d'inégalité sur les $L^k(c)$

Y

$y_s(c)$: Multiplicateur de Lagrange pour la contrainte sur les $B_s(c)$

Z

z : Fonction objectif

z_c : Coût du réseau

z_r : Revenu des connexions

CHAPITRE 1

Introduction

Les communications multipoint, impliquant plus de deux usagers simultanément, deviennent de plus en plus populaires. On cite à titre d'exemple des applications comme la vidéo-conférence, la diffusion de données audiovisuelle à des sites répartis dans le réseau, l'enseignement à distance et l'accès à des bases de données [98, 62, 23].

Un exemple de la popularité des communications multipoint est Mbone (*multi-cast backbone*) qui est un réseau virtuel permettant de réaliser des communications multipoint sur Internet [28, 2]. Bien que Mbone soit encore dans une phase expérimentale, un nombre considérable d'usagers l'utilisent dès aujourd'hui même si la qualité de service n'est pas encore au rendez-vous. En effet, lors de la diffusion de données multimédia, on constate que le son est de mauvaise qualité à cause des interruptions de la voix dues aux variations du délais de transmission, et les images sont rafraîchies à une fréquence très lente. Cela est dû à la largeur de bande limitée et à l'absence de garantie de qualité de service (qs)¹ dans Internet. D'autre part,

1. Nous utilisons cette notation afin de distinguer la qualité de service du niveau unités de données (qs) de celle du niveau appel (QS).

on peut s'attendre à une demande croissante en communications multipoint qui requièrent une grande largeur de bande avec une qs garantie, ce qui justifie le besoin d'introduire des techniques élaborées pour réaliser ce type de communications.

Le réseau numérique à intégration de services à large bande (RNIS-LB) utilisant le mode de transfert asynchrone (ATM: *Asynchronous Transfer Mode*) offre quant à lui une qualité de service garantie d'une part, et une grande largeur de bande, d'autre part. De plus, cette largeur de bande est flexible et peut être ajustée en fonction de la demande de l'utilisateur. Ces deux caractéristiques pourraient rendre envisageable un éventail d'applications qui requièrent à la fois une grande largeur de bande et une qs garantie. Hélas, l'ATM a une signalisation multipoint très lourde [7] et toutes les applications multipoint disponibles aujourd'hui utilisent Internet.

Aussi, en plus des communications point-à-point classiques, on voit apparaître des connexions multipoint qui peuvent être de trois types différents. On distingue le point-à-multipoint (diffusion de données multimédia), le multipoint-à-point (télémesures) ou encore, dans le cas général, le multipoint-à-multipoint (vidéo-conférence). Ces communications se différencient par la nature de leur trafic (variable ou constant), par leurs contraintes de qs, et par la largeur de bande requise.

Les communications multipoint sont encore nouvelles et on s'attend à ce qu'elles aient un impact important sur la gestion de ressources de télécommunications, notamment sur la synthèse de réseaux. En effet, certaines des techniques utilisées pour la gestion de ressources et la synthèse de réseaux traditionnels (communications point-à-point) [38, 58] risquent de ne plus être utilisables pour les communications multipoint. Par exemple, un appel point-à-multipoint d'une source donnée vers N destinations n'est pas équivalent à N appels point-à-point de cette source vers les N

destinations. En effet, on peut distinguer deux différences fondamentales 1) dans le cas d'un appel multipoint, le trafic est corrélé puisque plusieurs nœuds participent simultanément à une même communication et 2) en pratique, dans une connexion multipoint, les ressources sont partagées. En effet, on n'aura pas à allouer de la largeur de bande pour chacune des N destinations en utilisant N chemins distincts, mais plutôt on alloue de la largeur de bande pour la connexion multipoint en considérant que celle-ci est connectée à l'aide d'un arbre. Ceci est illustré par la figure 1.1 où on a N appels point-à-point connectés sur N chemins distincts, et par la figure 1.2 où on a un appel multipoint connecté à l'aide d'un arbre. Dans la figure 1.2, nous avons considéré que les nœuds sont capables de dupliquer l'information pour l'envoyer à plusieurs nœuds adjacents dans l'arbre. Ceci est réalisé par des commutateurs dupicateurs (voir définition dans la section 2.1). La duplication d'information permet surtout de faire des économies de ressources.

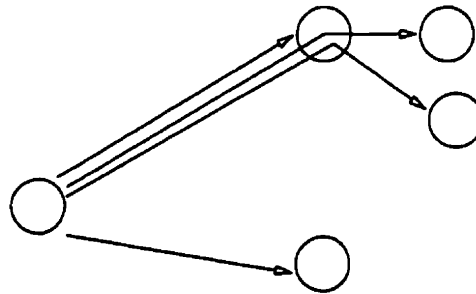


Figure 1.1: N appels point-à-point ($N = 4$ sur cette figure).

Comme on peut le voir sur cet exemple, avec l'introduction des communications multipoint, la gestion de ressources peut être affectée considérablement et les techniques utilisées dans le cas de communications point-à-point ne sont plus utilisables telles quelles. À ce jour, il n'y a dans la littérature aucune étude qui traite

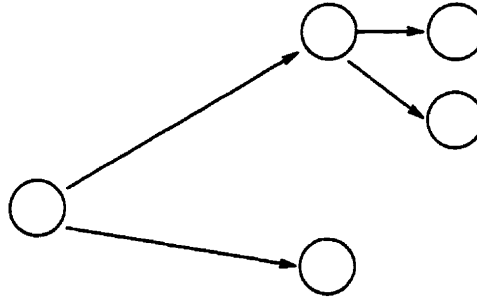


Figure 1.2: *Appel point-à-multipoint vers N destinations avec des commutateurs dupliateurs ($N = 4$).*

le problème de synthèse de réseaux avec communications multipoint. C'est l'une des principales motivations de notre projet de recherche. Nous voulons étudier l'impact de l'introduction des communications multipoint sur la gestion et les performances des réseaux de télécommunications et plus précisément, nous voulons étudier la synthèse de réseaux avec communications multipoint.

Avec l'introduction de communications multipoint, le problème de synthèse de réseaux, qui était relativement bien résolu pour le point-à-point, regagne de l'importance notamment à cause du volume considérable de trafic engendré par ce type de communications. D'une part, ce nouveau type de trafic exige une grande quantité de ressources, ce qui affecte considérablement le coût du réseau. D'autre part, le trafic multipoint peut potentiellement générer des revenus importants étant donné le grand nombre d'utilisateurs intéressés à ce type de communications. Tout cela fait en sorte que le problème de synthèse de réseaux doit être revu.

Le problème de synthèse de réseaux avec communications multipoint a été évoqué pour la première fois dans [40] où on donne une formulation mathématique et un cadre d'étude pour ce problème. Cet article soulève aussi des questions pertinentes

notamment, comment sélectionner les arbres qui permettent de maximiser le revenu, dans le contexte d'acheminement d'appels multipoint, ou de maximiser le revenu net du réseau dans le contexte de synthèse de réseaux. L'article souligne qu'il n'y a pas un choix naturel d'arbres pour connecter les appels multipoint, contrairement aux communications point-à-point où l'utilisation de l'arc direct est généralement le meilleur choix en termes d'utilisation de ressources.

1.1 Problématique

Le problème de synthèse de réseaux comprend deux sous-problèmes, l'acheminement des appels et le dimensionnement des artères de transmission. L'acheminement consiste à choisir les routes pour connecter les appels dans le but de maximiser le revenu généré par les connexions dans un réseau dont on connaît les capacités installées sur les arcs et les demandes de trafic. Le dimensionnement consiste à optimiser les capacités des artères de transmission dans le but de minimiser le coût du réseau pour des demandes de trafic données. Le problème de synthèse consiste alors à maximiser la valeur nette du réseau, soit la différence entre le revenu généré par les appels et le coût du réseau pour des demandes de trafic données.

Dans le contexte de réseaux avec communications point-à-point entre deux usagers, le problème de synthèse de réseaux a fait l'objet de nombreuses études [38, 58] qui ont montré que connecter les appels d'abord sur l'arc direct est généralement optimal en terme de revenu d'appels, dans le contexte d'acheminement, et en terme de revenu net, dans le contexte de synthèse de réseau.

Dans le contexte de réseaux avec communications multipoint, on ne parle plus de paire d'usagers voulant se connecter mais plutôt d'un ensemble d'usagers. On utilise

alors des arbres plutôt que des chemins pour connecter les appels. Contrairement au contexte de communications point-à-point, on ne sait pas comment sélectionner ces arbres de façon à optimiser le revenu des appels ou la valeur nette du réseau. Ainsi, le premier problème qui se pose est de sélectionner les arbres pour connecter les appels multipoint.

D'autre part, l'expérience montre qu'un réseau dimensionné pour du trafic point-à-point est peu sensible à des variations de l'intensité du trafic. Autrement dit, même si le trafic change par rapport aux valeurs pour lesquelles le réseau a été dimensionné, on est toujours capable de maintenir une qualité de service acceptable grâce en particulier aux techniques d'acheminement adaptatif, dans la mesure où ces variations ne sont pas trop fortes. Par contre, dans le cas du multipoint, on ne sait pas si un réseau dimensionné pour des demandes de trafic multipoint données garantirait une bonne qualité de service lorsque ces demandes de trafic changent. On doit alors étudier la sensibilité du réseau à la variation du trafic multipoint et éventuellement évaluer l'importance de la connaissance des demandes du trafic.

Finalement, dans les réseaux de télécommunications actuels, les applications sont de plus en plus gourmandes en termes de largeurs de bande. De plus, les exigences en largeur de bande peuvent varier considérablement d'une application à une autre par exemple, entre une communication téléphonique et la diffusion de données multimédia. On doit donc considérer les communications multipoint dans le contexte du multi-débit.

1.1.1 Le problème de sélection d'arbres

Afin de simplifier la discussion sur le problème de sélection d'arbres, nous considérons uniquement des communications point-à-multipoint. En effet, comme on le verra plus tard dans la section 2.4.2, ce type de communications permet aussi de réaliser des communications multipoint-à-multipoint et ce, en utilisant plusieurs communications point-à-multipoint, tel que recommandé par l'ATM Forum [7]. Dans la suite de ce chapitre, l'utilisation du terme multipoint se limite au cas point-à-multipoint.

L'un des points soulevés dans [40] est que pour les techniques d'acheminement utilisant un ensemble fixe pré-défini de routes, il n'y a pas de choix naturel d'arbres candidats dans le contexte du multipoint. Il nous faut donc trouver un critère pour la sélection d'arbres. Cette question comporte trois aspects:

1. Comment générer les arbres connaissant la source et les destinations? Les arbres le plus souvent utilisés sont les arbres de recouvrement de coût minimum, arbre du plus court chemin ainsi que les arbres de Steiner de coût minimum. Cependant, on ne sait pas quelle métrique utiliser pour générer ces arbres c.-à-d., quel est le coût qu'il faut associer à chaque arc du réseau. Il faut donc définir une métrique (coût des arcs) et des outils pour générer les arbres. Nous appelons ce problème le problème de *génération* d'arbres.
2. Une fois les arbres générés, comment construire des ensemble constituant les candidats pour l'acheminement? En d'autres termes, quels arbres, parmi les arbres qu'on a généré devrait-on sélectionner? Il faut définir un critère ainsi que des outils pour faire cette sélection. Ceci est le problème de *sélection* d'arbres. Nous donnons dans le chapitre 4 et dans [70, 72] une description de l'étude

que nous avons effectuée afin d'étudier le problème de sélection d'arbre dans le contexte d'acheminement et dans le chapitre 5 ainsi que dans [70, 71], une description de l'étude que nous avons effectuée dans le contexte de synthèse de réseaux.

3. Une fois les candidats connus, comment optimiser le *choix* d'arbres (l'acheminement) afin de maximiser le revenu? On suggère dans [40] que toutes autres choses étant par ailleurs égales, les grands arbres tels que les arbres de Steiner seraient des mauvais candidats à cause du grand nombre d'arcs qu'on y trouve. Nous avons besoin d'outils d'optimisation qui permettent de vérifier une telle conjecture. On peut utiliser des techniques d'optimisation classiques pour développer ces outils tel que décrit dans la section 3.4.

1.1.2 Point-à-point vs point-à-multipoint

Pour le problème de synthèse de réseau, on sait que pour des communications point-à-point, le réseau optimal (obtenu après dimensionnement) comprend les arcs directs entre presque toutes les paires de nœuds. Ainsi, dans le cas où on a plusieurs communications point-à-point, la topologie optimale est en général très fortement connectée [38] où la connectivité est mesurée par le nombre d'arcs dans le réseau, (les arcs représentent soit des liens physiques ou logiques).

Nous ne savons pas quelle serait la topologie optimale dans le cas où on a du trafic multipoint dans le réseau. En effet, dans la mesure où le trafic multipoint est connecté sur des arbres, ce qui est fondamentalement différent des arcs directs utilisés en point-à-point, on peut s'attendre à ce que la connectivité du réseau soit affectée par la présence du trafic multipoint. Nous voulons aussi savoir quelle serait

la topologie optimale du réseau dans le cas où les deux types de communications coexistent dans le réseau.

1.1.3 Sensibilité à la variation du trafic

En dimensionnant un réseau pour une matrice de trafic donnée, on obtient un réseau de *référence* qui est optimal pour une telle matrice de trafic. Un problème, peut-être aussi important que celui de la sélection d'arbres, est alors la sensibilité du réseau à la variation du trafic, soit par un changement dans les demandes de trafic, soit par un changement des ensembles de nœuds participants à des communication point-à-multipoint.

Le premier type de variation se manifeste lorsqu'on dimensionne le réseau pour des demandes de trafic estimées alors qu'en réalité, les demandes de trafic réelles sont sensiblement différentes lors de l'optimisation du réseau. Cette variation peut aussi se manifester si, après une période assez longue (quelques heures), les demandes de trafic changent et le réseau de référence peut alors ne plus être optimal pour ces nouvelles demandes de trafic.

Le deuxième type de variation se manifeste lorsqu'un certain nombre de nœuds, supposés prendre part dans les communications multipoint, ne sont plus là lors de l'établissement d'appels, alors que d'autres nœuds, non prévus, veulent se connecter.

La sensibilité aux variations de trafic est caractérisée par la dégradation possible de la qualité de service au niveau appel (QS) et/ou par une perte éventuelle de revenu. Ainsi, il faut quantifier cette sensibilité et éventuellement savoir de quelle manière il faut réagir aux deux types de variation de trafic. Cette étude nous permettra aussi d'évaluer l'importance de connaître à priori les matrices de trafic ainsi

que les différents membres participants aux communications multipoint. En effet, il est difficile en pratique de prévoir le trafic et on peut s'attendre à ce que le nombre de connexions multipoint dans le réseau soit très grand faisant en sorte qu'il est difficile de connaître le trafic à l'avance. Ainsi, l'hypothèse de connaissance des matrices de trafic n'est pas très réaliste mais on doit la considérer afin que nous puissions dégager les tendances liées au multipoint.

1.1.4 Multi-débit

Un modèle réaliste serait celui où on a un réseau multi-débit avec plusieurs types d'appels, point-à-point et point-à-multipoint, ce qui constitue l'objectif ultime de notre travail. Ceci nous permettra entre autre d'étudier le multipoint dans des réseaux à large bande de type ATM puisque des techniques utilisées dans les réseaux multi-débit sont applicables dans le contexte de réseaux de type ATM (car on utilise des circuits virtuels) en utilisant la notion de débit équivalent [37, 59, 25].

Nous voulons d'abord savoir si les résultats trouvés dans le contexte du mono-débit sont valides dans le contexte du multi-débit. Ensuite, nous voulons dégager les particularités liées à la coexistence du multi-débit et du multipoint sur la synthèse de réseaux.

1.2 Objectifs et contributions

Comme on vient de le voir dans la section 1.1, le problème de synthèse de réseau avec communications multipoint est très complexe. Nous avons présenté les points qu'on a jugé essentiels notamment, dans le contexte du mono-débit, les problèmes de choix optimal d'arbres, de sélection d'arbres et de sensibilité du réseau aux variations

de trafic. Ces problèmes doivent aussi être considérés dans le contexte du multi-débit. D'autres problèmes ont été considérés dont nous donnons un aperçu dans le chapitre 6.

1.2.1 Objectifs

En partant de la problématique décrite dans la section 1.1, on peut résumer les objectifs de ce travail de recherche comme suit:

1. Comprendre le problème de synthèse de réseaux avec communications multipoint et dégager les points importants qui doivent être étudiés en premier lieu.
2. Développer des algorithmes numériques qui permettent d'optimiser l'acheminement des appels d'une part, et le dimensionnement du réseau d'autre part.
3. Comprendre le problème de sélection d'arbres, évaluer son impact sur la synthèse de réseaux et trouver des critères simples pour cette sélection, dans le contexte d'acheminement d'une part et dans le contexte de dimensionnement d'autre part (on étudie d'abord les réseaux mono-débit).
4. Proposer un modèle pour le problème de sélection d'arbres et développer des procédures pour sa résolution.
5. Étudier la sensibilité du réseau à la variation du trafic multipoint en termes de perte de revenu et de qualité de service du niveau appel et éventuellement, si nécessaire, proposer un modèle adéquat qui tient compte de cette variation de trafic.

6. Étudier les points énumérés ci-dessus dans le cadre de grands réseaux multi-débit, incluant des communications point-à-point et point-à-multipoint. Ceci nous permettra d'étudier l'impact du multipoint sur la synthèse des réseaux à large bande de type ATM [76].

1.2.2 Contributions

À la lumière des objectifs cités dans la section 1.2.1, nous donnons dans cette sous-section les contributions réalisées dans le cadre de ce travail de recherche. Ces contributions ont fait l'objet d'une publication dans la conférence internationale *IEEE Globecom* [70], de deux articles soumis pour publications dans la revue spécialisée *IEEE/ACM Transactions on Networking* [72, 71] dont deux versions courtes ont été soumises pour publication à la conférence *ITC-16* [74, 73].

1. Dégagement des points essentiels à étudier dans le contexte de synthèse de réseaux supportant des communications multipoint.
2. Résolution des problèmes d'acheminement et de dimensionnement.
 - (a) Développement et mise en œuvre d'un algorithme basé sur la méthode du gradient projeté pour l'optimisation de l'acheminement.
 - (b) Développement et mise en œuvre d'un algorithme basé sur la méthode de relaxation Lagrangienne pour la résolution du problème de dimensionnement.
3. Établissement de critères de sélection d'arbres pour l'acheminement et pour le dimensionnement [70].

4. Développement de procédures pour la sélection d'arbres
 - (a) Deux algorithmes dans le contexte d'acheminement d'appels dont un basé sur une méthode exacte [72] et l'autre sur une heuristique [74].
 - (b) Un algorithme dans le contexte de synthèse de réseau basé sur une heuristique simple en tenant compte des communications point-à-multipoint et point-à-point [73, 71].
5. Étude du problème de sensibilité à la variation de trafic dans les contextes d'acheminement et de dimensionnement [71].
6. Généralisation des résultats dans le cas de réseaux multi-débit [72, 71].

1.3 Organisation de la thèse

Dans le chapitre 2, nous proposons une revue de la littérature qui décrit le multipoint avec les problèmes qui en découlent. Dans le chapitre 3, nous présentons les modèles d'acheminement et de synthèse de réseaux ainsi que les algorithmes que nous avons développés pour résoudre ces problèmes. Dans le chapitre 4, nous présentons la démarche que nous avons adoptée pour résoudre le problème de sélection d'arbres dans le contexte d'acheminement en présentant une méthode exacte et une méthode heuristique. Dans le chapitre 5, nous décrivons l'approche que nous proposons pour sélectionner les arbres dans les contextes de dimensionnement optimal et de synthèse de réseaux. Enfin, le chapitre 6 donne un résumé des contributions et un aperçu sur des travaux de recherche future qui peuvent constituer une suite à notre travail.

CHAPITRE 2

Revue de la littérature

Les communications multipoint deviennent de plus en plus populaires. Ceci a motivé un grand nombre de chercheurs à étudier les divers problèmes à surmonter afin d'offrir les communications multipoint de façon efficace. Parmi ces problèmes, comment réaliser le multipoint, comment gérer les ressources dans un tel contexte et comment offrir de telles communications avec une qualité de service garantie. Dans ce chapitre, nous proposons une revue de la littérature où nous décrivons brièvement le multipoint et ses problèmes les plus importants, en citant les travaux de recherche les plus pertinents qui traitent ces problèmes. Dans la section 2.1, nous exposons les caractéristiques du multipoint en général et les difficultés qui en découlent. Dans la section 2.3, nous décrivons spécifiquement le multipoint dans Internet et dans la section 2.4, nous décrivons le multipoint dans les réseaux de type ATM.

2.1 Le multipoint

Dans cette section, nous décrivons les caractéristiques du multipoint en général et les problèmes qui en découlent notamment, comment réaliser la duplication des unités de données, comment gérer les ressources (largeur de bande), comment réaliser l'adressage, comment réaliser la tarification et comment offrir de telles communications avec une qs garantie.

2.1.1 Commutateur duplicateur

Dans la littérature anglophone on emploie le terme *multicast* pour duplication. Le terme est souvent utilisé pour désigner l'opération qui consiste à dupliquer une unité de données (paquet ou cellule en ATM) provenant d'une source vers plusieurs destinations¹. Cette opération est réalisée à l'aide de commutateurs dupicateurs.

Un commutateur duplicateur permet de générer plusieurs copies d'une unité de donnée arrivant sur un port d'entrée vers plusieurs ports de sortie. Ceci permet d'envoyer simultanément plusieurs copies de cette unité de donnée à un certain nombre de destinations, sans avoir à établir une connexion point-à-point entre la source et chacune de ces destinations. Ceci permet aussi de réduire le trafic puisqu'il n'est pas nécessaire d'avoir N unités de données identiques sur un arc donné du réseau pour desservir N usagers. Il existe plusieurs techniques pour effectuer une telle opération de duplication.

1. On utilise ce terme aussi pour désigner la diffusion d'une information provenant d'une source vers un ensemble de destinations.

2.1.1.1 Techniques de duplication

Plusieurs techniques de duplication d'unités de données sont proposées dans la littérature. Nous allons décrire les techniques les plus populaires en commençant par les plus simples.

La duplication peut être réalisée à l'aide du recyclage d'une unité de donnée à l'intérieur du commutateur [92]. On utilise une boucle qui relie un port de sortie à un port d'entrée et à chaque cycle, une copie de cette unité est envoyée au port de sortie approprié. L'opération est répétée autant de fois qu'il y a de ports de sortie à partir desquels des copies doivent être envoyées vers des nœuds adjacents. Cette méthode de duplication n'est pas très équitable dans la mesure où les dernières copies générées peuvent arriver à destination très en retard par rapport aux premières copies. On propose alors [65] d'introduire un décalage cyclique dans l'ordre dans lequel les ports de sortie sont sélectionnés. Ce décalage est réalisé moyennant l'introduction d'une complexité matérielle non négligeable à l'intérieur du commutateur [65].

Une autre approche de duplication, utilisée surtout pour des commutateurs du type « Banyan » [26, 92], consiste à générer des copies en même temps que celles-ci sont acheminées à l'intérieur du commutateur [26]. L'en-tête est alors modifié afin d'indiquer si oui ou non une unité de donnée doit être dupliquée à un élément donné du commutateur. Cette méthode introduit un délai supplémentaire pour toutes les unités de données à cause du traitement de l'information sur la duplication lors de l'acheminement de ces unités de données et ce délai peut devenir très grand si plusieurs copies doivent être faites.

Enfin, une autre technique intéressante est proposée, surtout pour les commutateurs de type ATM à mémoires partagées. La duplication est réalisée par l'accès

à un bus auquel seront connectés seulement les ports de sortie par où des copies de cellules doivent quitter le commutateur [63]. Le seul inconvénient de ce type de commutateur est la limitation de la vitesse d'accès à la mémoire.

2.1.1.2 Serveur duplicateur

Dans la littérature [7, 23, 18] on propose l'utilisation de serveurs dupicateurs pour réaliser des connexion multipoint. Ce sont des commutateurs spécialisés qui sont les seuls commutateurs dans le réseau capables de faire la duplication d'unités de données. Toute source voulant diffuser des données à un certain nombre de destinations doit alors se connecter à un serveur duplicateur qui se charge de l'envoi de copies des unités de données, venant de la source, vers chacune des destinations. Cette idée est héritée de l'Internet où la notion de serveurs dupicateurs (équivalent à *multicast routers*) est très utilisée [23, 5, 13, 12, 94]. Cependant, il est aujourd'hui admis que pour les commutateurs de type ATM, la duplication de cellules ne coûte pas cher dans la mesure où la plupart des constructeurs l'ont déjà introduite dans leurs commutateurs. En pratique, nous pouvons considérer que tous les commutateurs d'un réseau de type ATM (et de plus en plus de routeurs Internet) sont dotés de la fonction de duplication de cellules.

2.1.2 Aspect dynamique des connexions

Une particularité spécifique aux communications multipoint est l'aspect dynamique des connexions. Cet aspect dynamique est caractérisé par la possibilité qu'un usager joigne et/ou quitte une communication multipoint donnée à tout moment pendant la durée de vie de celle-ci. L'arrivée et le départ des usagers peuvent être

gérés par la (les) source(s) de données ou par les destinations. Ainsi, la composition du groupe d'intervenants qui participent à une communication multipoint peut changer pendant la durée de vie de celle-ci. Plusieurs problèmes apparaissent alors, problèmes que l'on ne trouve pas dans le contexte de connexions point-à-point.

Le premier problème est celui d'identifier les membres prenant part à une connexion multipoint à un instant donné. Cette identification peut être difficile lorsque les usagers se connectent à leur initiative, ce qui est le cas d'Internet. Il faut alors introduire des mécanismes sophistiqués pour s'assurer que l'information concernant les membres connectés soit bien véhiculée à l'ensemble de tous les participants connectés (dont la composition peut changer). En pratique, un des nœuds est choisi comme responsable de recueillir et diffuser cette information. Il s'agit alors d'une approche centralisée où la liste des membres est gérée par un nœud qui est souvent la source de données [7]. Une approche décentralisée est aussi possible où chaque nœud doit maintenir à jour l'information sur les membres qui lui sont directement connectés. Un moyen connu sous le nom de *Mlisten* [2] qui permet d'identifier les différents membres d'un groupe multipoint dans Internet est décrit dans la section 2.3.

D'autre part, dans la mesure où les nœuds (commutateurs) membres d'une communication multipoint peuvent changer, l'arbre qui connecte ces nœuds peut ne pas être optimal en terme d'utilisation de ressources. En effet, il faudrait modifier l'arbre à chaque changement dans le groupe. Souvent, la modification de l'arbre n'est que partielle et ne concerne que les nœuds qui se sont connectés ou déconnectés rendant ainsi très difficile le maintien de l'optimalité de l'arbre en termes d'utilisation de ressources [54].

2.1.3 Qualité de service

Dans cette section, nous discutons de la qualité de service des niveaux unités de donnée (paquet ou cellule). Afin de la distinguer entre la qualité de service (QS) du niveau appel, nous noterons *qs* la qualité de service du niveau paquet.

La *qs* d'une connexion multipoint n'a pas encore été définie par les organismes de normalisation. La présence de plusieurs usagers à la fois, ayant chacun un équipement différent, fait en sorte qu'il est difficile de définir une même *qs* pour l'ensemble des usagers d'une même connexion. D'autre part, on ne sait pas lequel des membres d'une connexion multipoint doit déclarer les paramètres de *qs*. Par exemple, dans le cas de communications point-à-multipoint, la source pourrait déclarer les paramètres de *qs* de la connexion. Cela suppose que celle-ci connaît les besoins en *qs* de tous les usagers, ce qui semble difficile à implanter. Si, par contre, ce sont les destinations qui déclarent leur *qs*, alors, il faut introduire des mécanismes d'acceptation d'appel sophistiqués afin de pouvoir garantir la *qs* à chaque usager [90].

Il est de plus difficile de garantir une même *qs* à tous les membres d'une connexion multipoint dans la mesure où la position par rapport à la source n'est pas la même pour toutes les destinations. Nous avons alors un problème d'équité si la *qs* ne peut être garantie pour tous les membres. Par exemple, dans une conférence vidéo, il se peut qu'un usager, situé loin de la source et des autres membres, ne puisse pas obtenir les mêmes garanties de *qs* que les autres membres. On a donc besoin de spécifier ce que le réseau doit faire dans ce genre de situations.

De plus, les terminaux participant à une même communication multipoint peuvent avoir des caractéristiques différentes, notamment, la résolution graphique, ce qui

rend encore plus difficile l'attribution de la qs aux usagers utilisant de tels terminaux.

À ce jour, il n'y a aucune étude qui a été effectuée à cet égard. Puisqu'il n'y a aucune garantie de qs dans Internet, on a introduit des mécanismes de réservation de ressources [30] afin d'être en mesure d'offrir une certaine garantie de qs (pour le point-à-point et le multipoint). Nous donnons une description de tels mécanismes dans la section 2.3. En ATM par contre, c'est la source qui déclare la qs d'une connexion point-à-multipoint et tous les usagers auront ainsi la même qs [7]. Il n'y a cependant pas de définition de nouveaux paramètres qui tiennent compte de l'aspect multipoint notamment, la diversité des équipements des usagers et l'aspect dynamique des connexions. Dans la section 2.4, nous donnons des définitions relatives à la qs pour les connexions multipoint dans les réseaux de type ATM.

2.1.4 Adressage multipoint

Avec l'introduction de communications multipoint, l'adressage devient difficile à gérer dans la mesure où plusieurs participants prennent part à une même communication. En effet, si, dans le contexte du point-à-point, il est relativement facile d'identifier l'adresse d'une destination donnée, cela est moins évident dans le contexte où on a plusieurs destinations surtout si celles-ci peuvent changer durant les communications. De plus, dans la mesure où plusieurs usagers participent à une communication donnée et où on a plusieurs communications multipoint, l'espace d'adresse (contenu dans l'en-tête des unités de données) peut être rapidement épuisé si on ne gère pas l'espace d'adressage de façon efficace.

Afin de gérer efficacement l'espace d'adressage pour les communications multipoint, on a introduit la notion de *groupe* qui peut être identifié par une adresse unique. Les usagers participant à une communication multipoint font alors partie d'un même groupe et sont identifiés par une même adresse. Cette notion permet de simplifier l'identification d'une connexion multipoint dans le réseau surtout lorsque de nouveaux usagers veulent se connecter à une communication en cours.

La notion de groupe et d'adresse de groupe est très efficacement utilisée dans Internet [23, 5, 13]. Malheureusement, les réseaux de type ATM ne permettent pas d'utiliser efficacement cette notion de groupe dans la mesure où aucun champ dans l'en-tête des cellules n'a été prévu à cette fin. En effet, si, dans Internet, l'adresse utilisée dans l'en-tête des datagrammes ne change pas tout au long du parcours de ces derniers dans le réseau, dans les réseaux de type ATM, on utilise des numéros de voies logiques (VPI: *Virtual Path Identifier* et VCI: *Virtual Channel Identifier*) dans l'en-tête des cellules, qui peuvent changer si les cellules traversent des nœuds de transit (des commutateurs). Plus de détails sur l'adressage sont donnés pour Internet dans la section 2.3 et pour ATM dans la section 2.4.

2.1.5 Tarification

Le problème de tarification prend une nature différente dans le contexte de communications multipoint par rapport au contexte de communications point-à-point. En effet, dans la mesure où plusieurs usagers participent à la même connexion, la question qui se pose est alors quelle doit être la contribution de chacun de ces usagers dans le coût de la communication? Dans le contexte de communications point-à-point, en général, c'est le demandeur du service (l'appelant dans un réseau

de téléphone) qui paye le prix de la communication. On se demande alors comment le coût d'une communication multipoint sera partagé parmi les usagers.

Dans [46], on propose de faire une allocation équitable parmi les usagers. Il s'agit surtout de répartir équitablement les économies effectuées en utilisant un arbre unique pour connecter les usagers plutôt qu'un chemin pour chacun de ces usagers. Ce genre de problème peut être adéquatement étudié en utilisant la théorie des jeux coopératifs tel que suggéré dans [80].

Le coût peut être attribué en tenant compte du nombre de liens qui séparent un usager donné de la source [46]. On suppose que seuls les usagers connectés à la source auront une portion du coût qui leur est affecté et ce, en tenant compte du fait que la connexion d'un usager se fait à sa demande ce qui veut dire que la source n'est pas prise en compte dans la tarification. Il faut aussi tenir compte de la qualité de service qui peut varier d'un usager à l'autre [46]. Alors, la contribution d'un usager devrait augmenter ou diminuer selon que ses contraintes de qs sont très exigeantes ou peu exigeantes, respectivement.

2.2 Synthèse de réseaux

Pour concevoir un réseau de télécommunications, il est indispensable de définir les différents éléments qui constituent ce réseau. On peut alors donner une classification hiérarchique des différents niveaux d'un réseau de télécommunications. On distingue

- le réseau physique,
- le réseau de transmission,
- le réseau commuté et

- le réseau d'abonnés.

Le réseau physique est le niveau hiérarchique le plus bas sur lequel on construit le réseau de transmission et représente les liens physiques tels que câbles coaxiaux ou fibres optiques. Le réseau de transmission permet de raccorder les différents utilisateurs du réseau à l'aide de systèmes point-à-point. Il s'agit de systèmes de capacité fixe sur des périodes assez longues (plusieurs semaines) où la capacité est exprimée en terme de nombre de circuits, chaque circuit représentant une largeur de bande bien déterminée (par exemple, en téléphonie, un circuit correspond à 64 kbits/s). Les points de raccordement sont appelés multiplexeurs. En pratique, il n'est pas économique de raccorder tous les utilisateurs du réseau par des systèmes point-à-point et on fait appel aux techniques de multiplexage pour affecter de la capacité entre les utilisateurs [37]. Le réseau commuté, plus flexible que le réseau de transmission, est un niveau hiérarchique plus haut et est constitué des équipements (commutateurs) permettant d'affecter les ressources (largeur de bande) aux usagers sur des périodes qui varient selon le besoin des usagers (durées des communications). Enfin, le réseau d'abonnés est constitué des équipements qui permettent de raccorder les abonnés (usagers) au réseau à travers les commutateurs et les multiplexeurs.

Chaque niveau hiérarchique représente une demande pour le niveau hiérarchique inférieur. Par exemple, les appels des abonnés constituent la demande offerte au réseau commuté. La capacité requise entre les commutateurs du réseau commuté représente la demande pour le réseau de transmission. Enfin, la capacité des arcs du réseau de transmission constitue la demande à installer sur les liens physiques.

Dans ce travail de recherche, nous nous intéressons à la synthèse du réseau commuté. Ainsi, le terme réseau sera limité au réseau commuté.

Le problème de synthèse de réseaux constitue la motivation principale de ce travail de recherche. Ce problème comprend deux sous-problèmes, l'acheminement des appels et le dimensionnement des artères de transmission. L'acheminement consiste à choisir les routes pour connecter les appels dans le but de maximiser le revenu généré par les connexions dans un réseau donné, dont on connaît les capacités installées sur les arcs et pour des demandes de trafic données. Le dimensionnement consiste à optimiser les capacités des artères de transmission afin de réduire le coût du réseau. La synthèse du réseau consiste à maximiser la valeur nette du réseau, soit la différence entre le revenu généré par les appels et le coût de dimensionnement des artères de transmission pour des demandes de trafic données.

Dans cette section, nous allons nous attarder sur les problèmes de caractérisation des demandes de trafic d'une part et sur l'acheminement d'autre part dans la mesure où la plus grande complexité du problème de synthèse en dépend. L'objectif ici est de dégager, dans le contexte de synthèse de réseaux, les différences fondamentales entre le point-à-point et le multipoint. On peut en distinguer quatre:

1. Dans le contexte de connexions multipoint, il est très difficile de caractériser les demandes de trafic surtout à cause du très grand nombre de connexions et de participants dans chaque connexion multipoint et aussi à cause de l'aspect dynamique de ces connexions.
2. Pour connecter les appels multipoint, on utilise des arbres plutôt que des chemins. La génération d'arbres est nettement plus complexe que la génération de chemins. En effet, il n'est pas difficile d'imaginer que relier deux points dans un graphe donné est beaucoup plus simple que relier plusieurs points à la fois, le nombre de choix possible dans le premier cas étant beaucoup plus réduit

que dans le deuxième cas, augmentant considérablement la complexité des algorithmes de génération d'arbres. Nous donnons dans la section 2.2.2 une description détaillée des différents types d'arbres couramment utilisés dans les réseaux de communications.

3. Dans le contexte de communications multipoint, on ne sait pas quels arbres sélectionner pour connecter les appels multipoint de façon à maximiser le revenu des appels (acheminement) ou la valeur nette du réseau (synthèse). Dans le contexte du point-à-point, plusieurs études [38, 58] montrent que l'utilisation de l'arc direct (s'il est disponible) comme premier choix pour connecter les appels est optimale. La question de sélection d'arbres est primordiale et elle fait l'objet des chapitres 4 et 5 et des articles [70, 72, 71, 74, 73].
4. On ne sait pas quelles techniques d'acheminement utiliser pour connecter les appels multipoint. Une fois qu'on a sélectionné de bons arbres candidats pour connecter les appels, on ne sait pas quelle politique d'acheminement utiliser. Nous donnons dans la section 2.2.3 une description détaillée des techniques d'acheminement couramment utilisées dans le contexte du point-à-point et nous voulons savoir si on peut généraliser de telles techniques d'acheminement pour le multipoint.

2.2.1 Demandes de trafic

Dans les réseaux fonctionnant par commutation de circuits (téléphone), une demande de trafic désigne le taux d'arrivée d'appels d'un commutateur de départ vers un commutateur d'arrivée (en gros, il s'agit du nombre d'appels par unité de temps). Dans les réseaux de datagrammes (Internet), les demandes de trafic représentent le

nombre de paquets par unité de temps soit, le trafic total offert à un point quelconque du réseau (d'un nœud origine à un nœud destination). Dans les réseaux utilisant les circuits virtuels (ATM), on retrouve les deux définitions de demande de trafic c.-à-d., au niveau appel et au niveau paquet.

Dans le contexte de communications point-à-point et en commutation de circuit, il est convenable de modéliser les arrivées d'appel par un processus de Poisson. Exception faite de la téléphonie, où on a de bons modèles pour décrire le trafic [38], il est en général difficile de connaître l'intensité de ces demandes de trafic. En pratique, on utilise une prédiction de la demande [38] en se basant sur les utilisateurs potentiels des services de communication et sur leur comportement habituel (nombre moyen de demandes de connexions passées par exemple).

Dans Internet, dans la mesure où il n'y a pas de garantie de qs, on se contente de réagir à la congestion en augmentant la capacité là où cela devient nécessaire. Pour ce faire, on effectue des mesures pour déterminer la nature et l'intensité du trafic [16, 95]. On fait alors des statistiques sur la taille des datagrammes, la durée des connexions TCP (*Transport Control Protocol*)² et la composition du trafic (WWW (*World Wide Web*), FTP (*File Transfer Protocol*), Telnet etc). Ces mesures sont effectuées sur un lien donné du réseau et pour des périodes variant de 24 heures à 7 jours [95]. Si ces mesures indiquent que la capacité du lien ne suffit plus (trop de congestion), on l'augmente.

Le problème de caractérisation du trafic est cependant plus difficile dans le cas des réseaux de type ATM car on n'a presque aucune idée sur la nature du trafic ni sur son intensité. Il existe néanmoins plusieurs éléments pour caractériser le trafic

2. La notion de connexion n'existe pas dans Internet mais s'agissant de TCP, on utilise les numéros de ports source-destination qui permettent d'identifier une connexion.

notamment, les modèles de sources [32] et les descripteurs de trafic [53]. On distingue les types de trafic CBR (*Constant Bit Rate*), VBR (*Variable Bit Rate*), ABR (*Available Bit Rate*) et UBR (*Unspecified Bit Rate*). On définit alors les procédures d'acceptation d'appels (*CAC: Call Admission Control*). Selon les descripteurs de trafic, les paramètres de qualité de service de la source et les ressources disponibles dans le réseau, les procédures de CAC indiquent si oui ou non un appel venant de cette source peut être accepté [53, 32].

Le problème de caractérisation des demandes de trafic est encore plus difficile dans le contexte du multipoint essentiellement à cause du très grand nombre de participants dans les communications d'une part et à cause de l'aspect dynamique des connexions, d'autre part. En effet, dans ce cas, on ne sait pas comment caractériser le processus d'arrivée d'appels d'une source donnée vers toutes ses destinations. Il faudrait définir une demande pour chaque membre participant à une communication multipoint donnée si on veut caractériser adéquatement le processus d'arrivée d'appels. Dans le cas où les connexions ne sont pas dynamiques (ensemble fixe de destinations), on peut définir un ensemble source-destinations. La demande de trafic peut alors caractériser le trafic venant de la source vers ses destinations.

À date, il n'y a aucune étude qui a été effectuée pour adéquatement caractériser le trafic multipoint dans un réseau de téléphone public ou ATM. Pour Internet, on effectue des mesures de trafic en considérant les datagrammes offerts aux arcs du réseau [2, 68] ce qui donne une idée très limitée sur le trafic d'appels.

D'autre part, il faut souligner que la séparation entre les trafics en provenance des différents types de réseaux notamment, le réseau de téléphone public, Internet

et même ATM se fait de moins en moins. Cela va dans l'esprit d'intégration des services de communication. Par exemple, de nombreuses connexions Internet utilisent le réseau de téléphone public [89]. En effet, la plupart des abonnements résidentiels Internet passent par le réseau de téléphone public [78]. On envisage aussi faire des appels téléphoniques en passant directement par le réseau de données tel que Internet ou ATM [17]. De plus, on envisage faire des communications Internet via les réseaux de type ATM [5, 6] en allant jusqu'à intégrer la commutation ATM dans les routeurs IP (*Internet Protocol*) [100]. Cette intégration fait en sorte que l'évolution du trafic en termes de volume et de distribution dans n'importe quel des trois types de réseaux affecte le trafic sur les autres réseaux. Cela aura sans doute des répercussions non négligeables sur la planification de réseaux de télécommunications dans les années à venir.

Un autre aspect qui illustre la convergence des différents types de réseaux est l'utilisation de la notion d'identificateur de connexion (ou label) dans Internet version 6 [91]. Ce label simplifie l'acheminement des datagrammes (point-à-point ou multipoint) dans la mesure où les routeurs Internet l'utilisent pour offrir un service identique aux datagrammes d'une connexion TCP, par exemple, en imposant que ces datagrammes suivent tous le même chemin. Cela ressemble à la notion d'identificateur de circuit virtuel en ATM où les paquets d'une même connexion, appartenant à une même classe de service [50], suivent le même chemin.

D'autre part, dans les services de type *ABR* introduits en ATM, on fait du contrôle de congestion et la q.s ainsi que le débit d'une connexion donnée dépendent de l'état du réseau (pas de garantie) ce qui est très semblable au service TCP sur Internet [100].

2.2.2 Algorithmes de génération d'arbres

Dans le contexte du point-à-point, une route est définie par un chemin allant d'un nœud source vers un nœud destinataire et constitué d'une succession d'arcs, passant par des nœuds de transit, reliant une paire de nœuds.

Dans le contexte du multipoint, une connexion est réalisée à l'aide d'un arbre. Cet arbre est un sous-graphe du réseau qui doit connecter les nœuds participant à la communication multipoint. On peut aussi avoir d'autres nœuds dans l'arbre, en dehors de ceux qui participent à la communication. On a alors un arbre de Steiner au sens de la théorie des graphes [101]. Un exemple d'arbre de Steiner est donné dans la figure 2.1 où les nœuds participant à la communication et les liens constituant l'arbre sont représentés en gras.

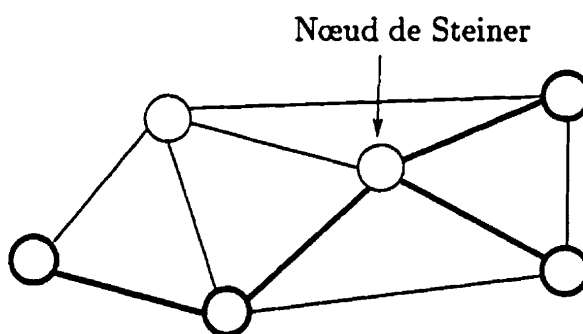


Figure 2.1: *Exemple d'arbre de Steiner.*

Le choix d'algorithme de génération d'arbres constitue un objectif primordial de notre travail de recherche. À priori, on ne sait pas quel type d'arbres est plus intéressant à utiliser pour connecter les appels multipoint. On peut distinguer deux critères pour choisir un algorithme plutôt qu'un autre: simplicité de l'algorithme et efficacité en termes de performances du réseau. Par exemple, un algorithme simple tel que celui de l'arbre des plus courts chemins qui minimise le coût (délai, argent,

distance, etc.) entre la source et toutes les destinations n'est pas nécessairement optimal en terme de revenu généré par les appels écoulés sur cet arbre et vice versa.

Nous partons de l'hypothèse que nous avons un graphe \mathcal{G} constitué d'un ensemble \mathcal{E} de $N = |\mathcal{E}|$ nœuds et d'un ensemble d'arcs \mathcal{S} . Nous supposons aussi que nous avons les coûts l_s de tous les arcs $s \in \mathcal{S}$. On veut connecter un sous ensemble $\mathcal{F} \subset \mathcal{E}$ de nœuds (nœuds obligatoires). Nous appelons alors $\mathcal{G}' \subseteq \mathcal{G}$, le sous-graphe construit pour connecter ces nœuds.

Nous définissons le poids P_l d'un arbre comme étant la somme des coûts l_s des arcs contenus dans l'arbre c.-à-d.,

$$P_l = \sum_{s \in \mathcal{G}'} l_s.$$

On distingue deux catégories d'algorithmes de génération d'arbres. Les algorithmes *statiques*, c.-à-d., où on génère un arbre pour un ensemble connu de nœuds. L'autre catégorie d'algorithmes sont ceux utilisés pour des ensembles de nœuds qui peuvent changer. L'arbre est alors modifié en conséquence par l'ajout ou la suppression d'arcs afin de connecter tous les nœuds participants [54]. On parle alors d'algorithmes *dynamiques*.

2.2.2.1 Algorithmes statiques

Ces algorithmes génèrent des arbres optimaux selon un critère bien déterminé, pour un ensemble de nœuds fixe, connu à l'avance (la génération d'arbres se fait hors ligne). Le critère est souvent lié au poids P_l de l'arbre que l'on minimise.

Plusieurs algorithmes sont proposés pour générer des arbres pour les communications multipoint [60, 69, 98]. Les algorithmes les plus populaires sont les suivants:

- Inondation (*Flooding*): c'est la méthode la plus simple pour diffuser les données dans un réseau. Il s'agit d'envoyer des copies d'une unité de données à tous les nœuds du réseau, même à ceux qui ne devraient pas recevoir ces données. Dans ce cas, \mathcal{G}' connecte tous les nœuds de \mathcal{E} mais peut contenir des cycles. Afin d'éviter d'avoir des cycles, on définit pour chaque nœud n du réseau un nœud père qui est le seul nœud à pouvoir envoyer des messages à ce nœud n ; et pour chaque nœud père on définit un ensemble de nœuds fils. Un exemple d'algorithme d'inondation populaire est RPF (*Reserve Path Forwarding*) [19]. Dans cet algorithme, chaque nœud père envoie une copie d'un message (unité de donnée), qu'il reçoit pour la première fois, à tous ses nœuds fils [21]. La complexité d'un tel algorithme est $O(N)$.
- Arbre de recouvrement: un arbre de recouvrement est un sous-graphe \mathcal{G}' ne contenant aucun cycle et incluant seulement les nœuds de \mathcal{F} (ceux qu'on veut connecter). La complexité d'un tel algorithme est $O(N)$.
- Arbre de recouvrement de coût minimum (*MST: Minimum Spanning Tree*): un arbre de recouvrement de coût minimum est un arbre de recouvrement dont le poids P_l est minimal. La complexité du calcul est $O(N^2)$.
- Arbre des plus courts chemins: construire un arbre des plus courts chemins consiste à trouver, pour un nœud $n \in \mathcal{F}$ donné, les chemins les plus courts au sens des coûts l , pour aller de ce nœud vers tous les autres nœuds de \mathcal{F} . n peut

être considéré comme la racine de l'arbre. La complexité de cet algorithme est aussi $O(N^2)$.

- Arbre de Steiner: le problème de recherche de l'arbre de Steiner dans un graphe consiste à trouver un sous graphe \mathcal{G}' , incluant tous les nœuds de \mathcal{F} (nœuds obligatoires), et éventuellement d'autres nœuds de \mathcal{E} [101].
- Arbre de Steiner de coût minimum: il s'agit d'un arbre de Steiner dont le poids P_l est minimum (ce problème est NP difficile [34]).

2.2.2.2 Algorithmes dynamiques

Plutôt que de générer un arbre optimal à l'aide d'un algorithme statique qui nécessite la génération complète d'un nouvel arbre à chaque fois que la composition de \mathcal{F} est modifiée, on propose des algorithmes où les arbres sont modifiés et mis à jour en fonction des usagers connectés [24]. De ces algorithmes, les plus connus sont les suivants:

- Inondation et Émondage (*Flood and Prune*): dans l'algorithme d'inondation, certains nœuds reçoivent inutilement des données alors qu'ils ne font pas partie de \mathcal{F} . Afin de remédier à cela, il a été proposé d'émonder l'arbre généré [97] par inondation afin d'éliminer les branches inutiles, celles qui ne contiennent aucun nœud de \mathcal{F} . L'opération d'inondation et d'émondage s'effectue périodiquement et tient compte de la nouvelle composition de \mathcal{F} . Une application est l'algorithme populaire proposé pour Internet connu sous le nom de PIM (*Protocol Independent Multicast*) [21] où, pour chaque source voulant diffuser des données, un arbre est généré par inondation et émondage (voir section 2.3.4 pour plus de détails sur l'algorithme PIM).

- Arbre avec centre [9, 15]: on construit un arbre des plus courts chemins entre le centre et chaque destination. Le centre n'est pas nécessairement une source mais on optimise son choix dans \mathcal{E} afin de minimiser les coûts des chemins entre ce centre et tous les nœuds de \mathcal{F} . Toutes les sources doivent passer par le centre afin de diffuser les données aux autres membres. Les membres utilisent donc un arbre unique (*Shared Tree*) pour communiquer entre eux. On peut avoir plusieurs centres dans un seul arbre. Pour chaque destination, l'arbre comprend alors le plus court chemin entre cette destination et le centre le plus proche [9]. Un exemple d'algorithme populaire pour générer ce type d'arbres est CBT (*Core Based Trees*) [9] introduit pour Internet (voir section 2.3.4 pour plus de détails sur l'algorithme CBT).

2.2.2.3 Comparaison des algorithmes de génération d'arbres

En ce qui concerne l'arbre de recouvrement, il s'agit d'un algorithme très simple. Cependant, il n'est pas toujours possible de générer un arbre de recouvrement car cela dépend de la connectivité du graphe du réseau. Nous avons alors besoin de générer un arbre de Steiner [101, 48]. Le problème de génération d'arbres de Steiner est NP difficile [34] et on a souvent recours à des heuristiques pour trouver de tels arbres [84, 93, 36, 98, 61, 44, 1].

Comme tous les arbres de poids minimum, l'arbre des plus courts chemins dépend de la métrique utilisée. L'arbre ainsi généré peut ne pas être optimal pour d'autres métriques tel que le coût des ressources utilisées. Ainsi, on ne sait pas quelle métrique utiliser pour générer les arbres afin de maximiser le revenu ou de minimiser le coût du réseau. Dans la littérature, l'optimalité des arbres est souvent liée au délai de

livraison des unités de données de la source vers les destinations [82, 88, 21] ou à des coûts d'arcs plus ou moins clairement définis [49, 86].

En ce qui concerne l'algorithme d'inondation, il présente un gaspillage non négligeable de ressources dans la mesure où certaines branches peuvent ne contenir aucun nœud destinataire. L'algorithme d'émondage permet de pallier cet inconvénient. Cependant, l'algorithme d'inondation et émondage a un problème de gaspillage de ressources dans le cas où on a plusieurs source, dans la mesure où il faut générer un arbre par source.

Enfin, l'arbre avec centre est très populaire à cause de sa souplesse lorsque les connexions sont dynamiques. De plus, il utilise un arbre partagé et on a donc un seul arbre peu importe le nombre de sources. Cependant, le problème de choix du centre est NP difficile [23]. Un autre inconvénient de l'arbre avec centre est l'inéquité de l'arbre dans la mesure où les nœuds proches du centre peuvent être privilégiés par rapport aux autres. Nous avons aussi le problème de concentration du trafic autour du centre ce qui peut créer une congestion.

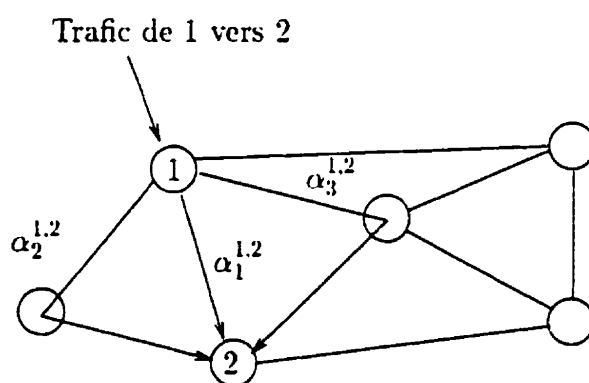
2.2.3 Techniques d'acheminement d'appels

Les techniques d'acheminement que nous décrivons ici sont couramment utilisées pour acheminer les *appels* point-à-point en commutation de circuits. Nous voulons entre autre savoir si de telles techniques pourraient être utilisées dans le contexte du multipoint.

2.2.3.1 Partage de charge

C'est la technique d'acheminement la plus simple. Cependant, elle n'est pas utilisée en pratique à cause de son inefficacité. On la retient surtout pour les avantages qu'elle présente en termes de simplicité et intérêt théorique.

Pour chaque paire de nœuds (i, j) , on définit un ensemble de K chemins $C^{i,j}$ qui peuvent être utilisés pour acheminer ces appels. L'objectif est alors d'optimiser le partage de charge soit, trouver les valeurs optimales des coefficients $\alpha_k^{i,j}$ représentant la fraction d'appels offerts au chemin (i, j, k) . En pratique, on effectue un tirage aléatoire selon ces coefficients, pour savoir sur quel chemin envoyer l'appel (voir figure 2.2). La faiblesse de cette méthode vient du fait qu'un appel peut être perdu (si le chemin auquel il est offert est bloqué), alors qu'il aurait pu être connecté sur un autre chemin libre, puisqu'il n'y a pas de deuxième tentative en cas de blocage.



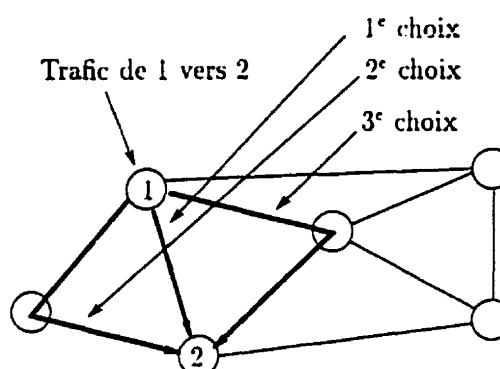
$$K = 3 \text{ chemins pour connecter les nœuds 1 et 2}$$

$$\alpha_1^{1,2} + \alpha_2^{1,2} + \alpha_3^{1,2} = 1$$

Figure 2.2: Exemple d'acheminement avec partage de charge.

2.2.3.2 Débordement

Comme dans le partage de charge, pour chaque paire de nœuds (i, j) , on définit un ensemble de chemins $C^{i,j}$ qui peuvent être utilisés pour acheminer les appels. Cette fois, ces chemins sont ordonnés. À son arrivée, un appel est offert au chemin de premier choix. Si ce dernier est bloqué, alors, l'appel est offert au chemin suivant et ainsi de suite, jusqu'à épuisement de tous les chemins de débordement (voir figure 2.3).



$K = 3$ chemins avec deux débordements pour connecter les nœuds 1 et 2

Figure 2.3: Exemple d'acheminement avec débordement.

Nous avons plusieurs façons de mettre en oeuvre le débordement [38]. La technique la plus souvent utilisée est appelée commande pas-à-pas ou encore *Sequential Office Control (SOC)*. Un appel bloqué à un nœud intermédiaire est perdu, c.-à-d. que la décision est prise uniquement à partir de l'état des liaisons adjacentes au commutateur où se trouve la commande (pas de possibilité de retour en arrière). Une autre technique de commande de bout-en-bout ou encore *Originating Office*

Control (OOC) permet de prendre une décision à partir de l'état complet du chemin (possibilité de retour en arrière). Ainsi, pour un ensemble de chemins donné, la condition de perte d'appel varie selon la commande utilisée.

2.2.3.3 Acheminement DNHR

L'acheminement DNHR (*Dynamic NonHierarchical Routing*) a été proposé par AT&T dans les années 80 [38]. On définit un ensemble de périodes pendant lesquelles on estime que les demandes de trafic ne changent pas. Au début, on a découpé la journée en 24 périodes d'une heure pendant lesquelles chaque paire origine-destination utilisait un acheminement non hiérarchique à plusieurs débordements avec transit. L'ensemble de chemins de débordement peut varier d'une période à l'autre. Nous avons donc un acheminement dynamique. Cependant, cet acheminement ne tient pas compte de l'état du réseau (congestion). Le DNHR n'est donc pas adaptatif.

2.2.3.4 Acheminements adaptatifs

Dans l'acheminement par partage de charge, l'inefficacité était due à la possibilité de perdre un appel alors qu'il aurait pu être connecté sur un deuxième chemin. On peut, au lieu de faire un tirage aléatoire, choisir un chemin d'une façon plus intelligente par exemple, en ajustant les coefficients de partage de charge selon l'état du réseau.

Un acheminement connu sous le nom de DCR [87] (*Dynamically Controlled Routing*) choisit le chemin de transit dont la capacité disponible est la plus grande possible. Un appel de i vers j est d'abord offert à l'arc direct (i, j) . Si cet arc est bloqué, on choisit le chemin de débordement k de façon aléatoire mais dont la probabilité de sélection $\alpha_k^{i,j}$ est calculée à partir de la capacité résiduelle $C_k^{i,j}$ [37] du

chemin. En gros, $C_k^{i,j}$ est le minimum des capacités résiduelles des arcs du chemin (obtenues en mesurant le nombre de circuits installés sur les arcs dont on retranche le nombre de circuits occupés) [37]. On a

$$\alpha_k^{i,j} = \frac{C_k^{i,j}}{\sum_l C_l^{i,j}}.$$

2.2.3.5 Acheminement d'appels multipoint

Dans la littérature, très peu d'études ont traité le problème d'acheminement d'appels multipoint en utilisant les techniques d'acheminement décrites ci-dessus. C'est l'une des principales motivations de ce travail de recherche. Par exemple, dans [40], on propose une généralisation de l'acheminement par partage de charge pour le multipoint avec l'hypothèse que toutes les destinations se connectent et se déconnectent en même temps.

On voit tout de suite la difficulté d'appliquer les techniques d'acheminement décrites ci-dessus dans le cas où les membres participant à une communication donnée peuvent changer puisqu'alors, on ne peut plus identifier un appel par une paire source-destination(s). Il faudrait plutôt considérer chaque destination à part, ce qui exigerait des techniques d'acheminement très sophistiquées.

2.2.4 Acheminement de paquets

Dans le cas de réseaux de datagrammes comme Internet, ce sont ces *datagrammes* qui sont acheminés plutôt que les appels. Chaque datagramme d'une même communication peut suivre un chemin distinct. Dans les réseaux avec circuits virtuels comme ATM, on achemine plutôt les circuits virtuels (un VP ou un VC) et tous les paquets d'une même connexion suivront alors le même chemin. Les réseaux utilisant

les circuits virtuels sont en fait une solution intermédiaire entre la commutation de circuits et la commutation de datagrammes. D'une part, cela permet de simplifier l'acheminement dans la mesure où on achemine un circuit virtuel à l'établissement de la connexion seulement (comme en commutation de circuits) et d'autre part, cela permet de ne pas gaspiller les ressources dans la mesure où plusieurs circuits virtuels puissent emprunter le même chemin physique faisant en sorte que les paquets de différentes connexions peuvent être acheminés sur le même chemin physique (comme pour les datagramme) grâce au multiplexage statistique. Le gain de multiplexage est surtout dû à la nature sporadique du trafic de données [10].

Nous donnons dans la section 2.3 une description des techniques d'acheminement multipoint utilisées dans Internet et dans la section 2.4 celles utilisées dans les réseaux de type ATM. On donne aussi dans la section 2.5 une description des techniques proposées pour acheminer le trafic multipoint d'Internet par les réseaux de type ATM.

2.3 Le multipoint dans Internet

Dans les années 90, les communications multipoint dans Internet ont progressé considérablement. Il y a eu de nombreux essais pour tester le fonctionnement de ce type de communications [2]. Dans cette section nous donnons les éléments particuliers à ce type de communications dans Internet.

2.3.1 Le réseau Mbone

Mbone est un réseau virtuel au dessus de Internet qui s'occupe de tous les aspects du trafic multipoint comme l'acheminement, l'adressage et la signalisation [28]. Il

faut noter que Mbone est un réseau expérimental mais qu'il est sur le point de devenir le support multipoint pour Internet.

Mbone est constitué d'un ensemble de sous-réseaux qu'on appelle *îles*. À chaque île, on associe un *serveur dupicateur* sur lequel est exécuté un programme appelé *Mrouted*. Ces serveurs sont tous connectés entre eux à l'aide de liens point-à-point appelés *tunnels* (les tunnels sont routés sur les liens physiques du réseau). Ceci est illustré par la figure 2.4.

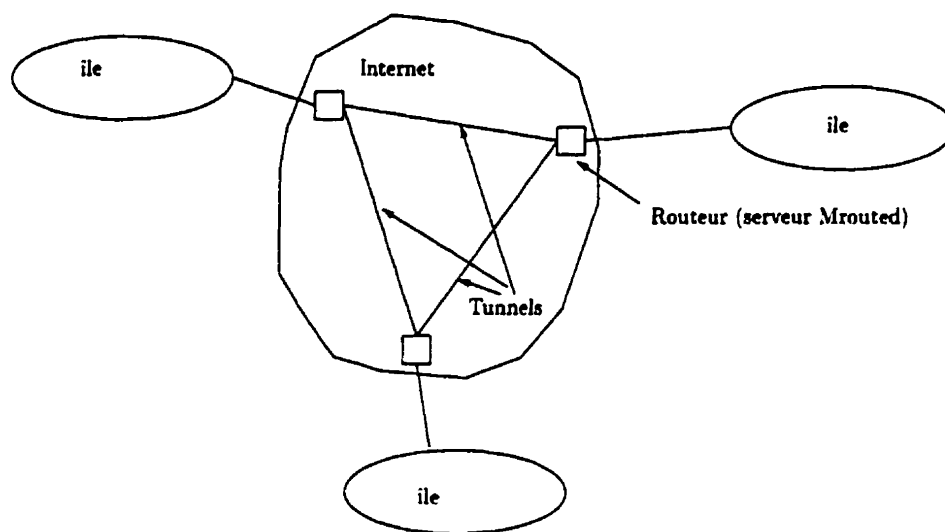


Figure 2.4: *Topologie simplifiée de Mbone.*

La procédure avec laquelle les données sont envoyées est la suivante: un paquet (datagramme) est envoyé d'un réseau local (appartenant à une île) à un serveur Mrouted. Ce paquet est destiné à un certain groupe de destinations. La duplication de paquets se fait par les routeurs Mbone. Le programme Mrouted se charge alors de déterminer sur quel(s) tunnel(s) le paquet doit être envoyé. Des copies de ce paquet arrivent alors à d'autres serveurs qui eux vont effectuer la même opération jusqu'à ce que les destinations reçoivent chacune une copie du paquet.

2.3.2 Protocole UDP

Le trafic (du niveau paquet) sur Mbone est acheminé en utilisant un protocole appelé UDP (*User Datagram Protocol*), plutôt que TCP. Contrairement à TCP, UDP ne fournit aucun contrôle d'erreur ni contrôle de débit (pas de voie de retour) et il n'y a donc pas de garantie de qualité de service (qs). UDP a été adopté car les procédures de contrôle de flux standard sont trop lentes pour le trafic multipoint [28]. En effet, le délai dû au contrôle d'erreurs de TCP (retransmissions) est inacceptable pour du trafic multipoint car en général, ce type de trafic est sensible au délai de transmission (trafic vidéo, temps réel etc.). Ainsi, avec le protocole UDP, on ne parle plus de connexion comme c'est le cas pour TCP mais plutôt de *sessions* Internet.

Pour les applications multipoint sensibles à la perte d'information, certains chercheurs proposent l'utilisation de codes correcteurs d'erreurs (*FEC: Forward Error Control*) [81] où on ajoute des bits de parité aux datagrammes transmis afin de corriger les erreurs de transmission à la réception (la correction d'erreurs n'est possible que si le nombre de bits erronés n'est pas plus grand que le nombre de bits de parité reçus [81]). Cependant, avec cette approche, on a un gaspillage de largeur de bande dû aux bits de parité transmis avec chaque datagramme.

2.3.3 Protocole IGMP

Le protocole IGMP (*Internet Group Management Protocol*) a été introduit pour permettre à différents groupes multipoint de joindre ou de quitter des sessions multipoint [20]. Un groupe multipoint désigne un ensemble d'utilisateurs, généralement appartenant à un même réseau local.

Un hôte est désigné pour représenter un groupe donné. Cet hôte peut représenter plusieurs groupes à la fois et n'est pas nécessairement membre d'un groupe multipoint auquel cas, l'hôte ne reçoit pas les données échangées par les usagers. Les hôtes s'échangent de l'information concernant les membres qu'ils représentent. Cette information sert entre autre à connaître les différents groupes participant à une session multipoint à un instant donné. Ces groupes peuvent changer en tout temps pendant la session et il n'y a aucune restriction sur leur nombre ni leurs positions. Dans [31], on a introduit une procédure qui permet aux réseaux locaux d'élire leurs hôtes.

Après identification de tous les membres, Mbone se charge alors de diffuser les datagrammes à tous les groupes qui sont membres d'une session multipoint, en utilisant un acheminement approprié.

2.3.4 Acheminement

L'acheminement le plus populaire dans Internet se fait à l'aide de la procédure OSPF (*Open Shortest Path First*). Comme son nom l'indique, cette technique utilise un algorithme du plus court chemin pour acheminer les datagrammes. Une extension à OSPF a été faite pour le trafic multipoint. Celle-ci porte le nom de MOSPF pour *MulticastOSPF* [79].

Pour un datagramme donné, MOSPF utilise l'algorithme de Dijkstra pour trouver l'arbre des plus courts chemins, en utilisant les coûts spécifiés par le type de service (TOS: *Type of Service*). Ces services spécifient la métrique utilisée pour générer les arbres. Nous avons les services

- à délai minimum,

- à débit maximum,
- de coût minimum (en dollars) et
- normaux.

MOSPF utilise le protocole IGMP afin d'identifier les différents groupes qui sont membres d'une communication multipoint donnée. MOSPF se charge alors de livrer les datagrammes à chaque groupe, en envoyant les datagrammes sur le chemin le plus court de la source du datagramme vers les destinations. Il faut noter que dans MOSPF, les nœuds de terminaison (sources et destinations) ne sont pas les usagers situés dans les réseaux locaux mais ce sont plutôt les serveurs dupicateurs (routeurs Internet).

Afin d'acheminer les datagrammes en tenant compte de l'aspect dynamique des sessions multipoint, on a proposé les algorithmes PIM [21] et CBT [9] qui sont en fait des algorithmes de génération d'arbres dynamiques (voir section 2.2.2.1).

L'algorithme PIM consiste à générer un arbre des plus courts chemins pour chaque source d'un groupe multipoint donné par inondation et émondage. L'opération d'émondage s'effectue après que les serveurs dupicateurs, ayant reçu des données relatives aux membres du groupe multipoint et qui n'ont aucun membre du groupe multipoint dans leur réseau local, envoient aux sources de l'information (dans des datagrammes appropriés) afin que celles-ci suppriment les branches qui ne contiennent aucun membre du groupe des arbres générés par inondation [21].

D'autre part, l'algorithme CBT met à jour un arbre partagé pour chaque groupe multipoint. Cet arbre est un arbre des plus courts chemins généré à un centre pré-défini. On peut avoir plusieurs centres dans un seul arbre. En général, on a un centre par groupe multipoint [15]. Pour chaque destination, l'arbre comprend le

plus court chemin entre elle et le centre le plus proche [9]. Ce type d'arbre permet une grande souplesse pour l'ajout et la suppression de nœuds dans l'arbre dans la mesure où l'arbre généré dépend de la position du (des) centre(s) et non pas de la (des) source(s).

Aujourd'hui, on ne sait pas lequel des deux algorithmes PIM et CBT est plus efficace pour Internet en terme de performances du réseau. L'algorithme CBT génère un arbre unique, indépendamment du nombre de sources. Cependant, à cause du fait que les paquets n'empruntent pas le chemin le plus court de la source vers les destinations, le délai de transmission de paquets de bout en bout est plus grand avec l'algorithme CBT [11]. D'autre part, avec l'algorithme PIM, on a un gaspillage de ressources (largeur de bande) lorsque le nombre de sources est grand [11] à cause du nombre considérable d'arbres qui en résulte.

2.3.5 Qualité de service

En pratique, Internet n'offre pas une garantie de qualité de service (qs) au niveau de la couche réseau. Par exemple, la perte de paquets en cas de congestion ou erreurs de transmission est prise en compte par TCP à l'aide de procédures de retransmission que l'on considère inefficace dans le cas de trafic multipoint [28].

Afin d'offrir une garantie de qualité de service pour les applications qui utilisent Internet, on a proposé un protocole de réservation de ressources (*RSVP:Resource Reservation Protocol*) qui sert pour le trafic point-à-point de même que multipoint [14, 99]. RSVP permet aux applications de spécifier leurs besoins en termes de qualité de service et de donner une description de la nature du trafic engendré par ces applications. RSVP permet aux usagers participant à une même connexion

d'avoir des qs différentes et permet aussi aux usagers de changer leur qs durant la communication [30].

Une autre façon d'offrir une garantie de qs est de véhiculer les datagrammes Internet sur un réseau de type ATM qui, lui, offre une garantie de qs. Une description de cette approche est donnée dans la section 2.5.

2.3.6 Adressage

Le problème d'adressage du trafic multipoint est résolu dans Internet. Nous avons quatre classes d'adresses, A, B, C et D. En gros, d'une classe à l'autre, c'est l'intervalle des valeurs possibles qui change. La classe D est réservée au trafic multipoint [83].

Dans Internet, on définit la notion d'adresse de groupe qui identifie tous les membres d'un groupe participant à une communication multipoint. Tous les membres du groupe envoient les datagrammes en utilisant cette adresse pour désigner les destinations de leurs datagrammes. Ainsi, un membre donné peut ne pas connaître les autres membres du groupe. Ce sont donc les destinations qui contrôlent leur appartenance aux groupes. On peut voir ici le caractère non connecté (*connectionless*) du multipoint dans Internet.

2.3.7 Identification des membres

Un outil informatique connu sous le nom de *Mlisten* [2] permet de faire une collecte sur les différents membres d'un groupe multipoint donné afin de maintenir à jour une liste sur ces membres. Comme Mrouted, Mlisten est un programme exécuté sur un serveur. Mlisten peut joindre une communication multipoint et reconnaître

les membres du groupe en lisant l'adresse IP des paquets envoyés au groupe. L'inconvénient de cette approche est qu'elle ne fonctionne pas adéquatement si des paquets sont perdus [2].

2.3.8 Résumé

À date, Internet est le seul réseau de données accessible au public qui fournit des services de communications multipoint sur une grande étendue géographique. Des exemples pratiques ne manquent pas. On cite le protocole *SDP: Session Description Protocol* [45] qui permet d'identifier les sessions multimédia en cours sur Internet comme la diffusion de conférences internationales, de forum de discussions etc³.

La mauvaise qs et la largeur de bande limitée constituent l'handicap majeur du multipoint sur Internet. Beaucoup d'améliorations doivent être introduites notamment pour offrir plus de largeur de bande et une qs garantie. Cela exige une gestion efficace des ressources faute de quoi, le coût du réseau peut être très élevé et par conséquent, une communication multipoint avec qs garantie sur Internet pourrait s'avérer extrêmement coûteuse.

2.4 Le multipoint dans l'ATM

Une première spécification de communications multipoint est donnée par l'ATM Forum [7] qui spécifie essentiellement la signalisation et les procédures d'établissement de connexions multipoint. On trouve aussi des définitions de divers types

3. Il existe un service de conférences téléphoniques sur le réseau de téléphone public mais puisque ce service fait appel à plusieurs connexions point-à-point, nous ne le considérons pas comme étant un service de communications multipoint.

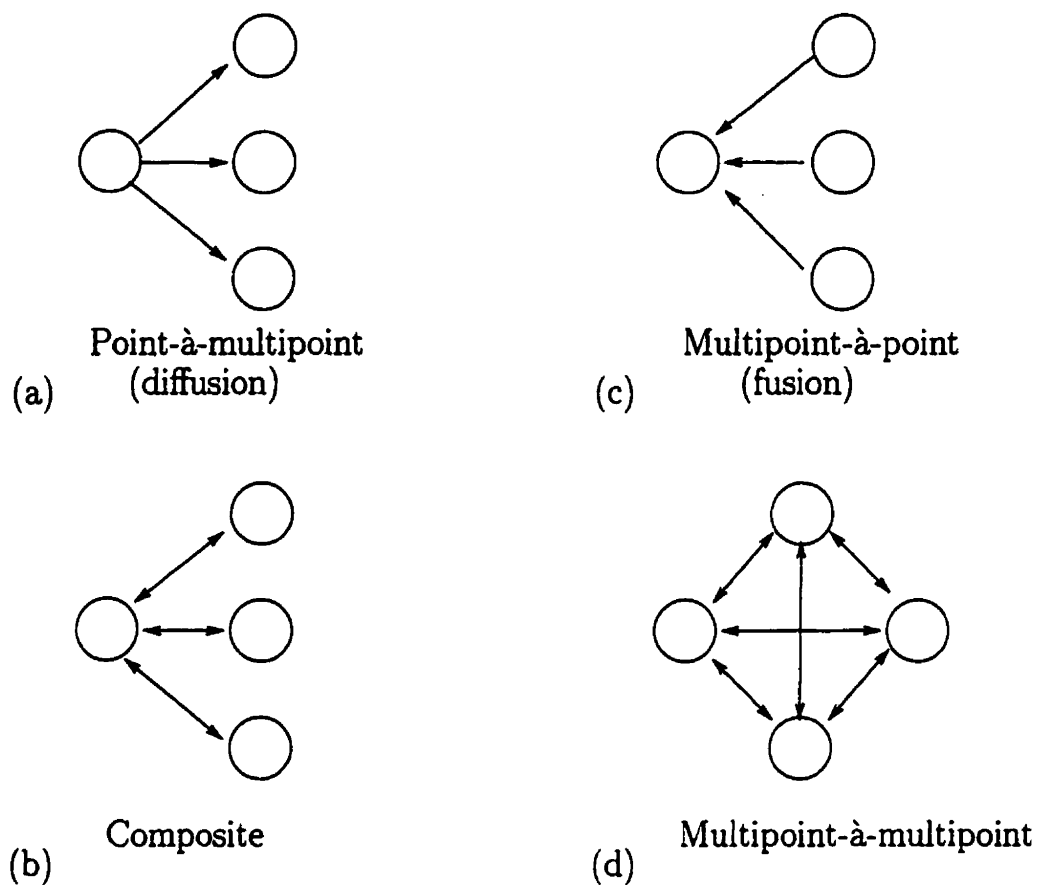


Figure 2.5: *Types de communications multipoint.*

de communications multipoint dans les recommandations de l'UIT-T (Union Internationale des Télécommunications-secteur de standardisation des Télécommunications) [52].

2.4.1 Définitions de l'UIT-T

Dans [52], on peut distinguer quatre types de connexions multipoint.

2.4.1.1 Point-à-multipoint

Ce type de communications est aussi connu sous le nom de diffusion (*broadcast*). Le trafic est unidirectionnel d'une source vers un ensemble de destinations (voir figure 2.5(a)) par exemple, la diffusion de données audiovisuelles.

2.4.1.2 Multipoint-à-point

Aussi connu sous le nom de fusion (*merge*). Le trafic est unidirectionnel d'un ensemble de sources vers une destination commune (voir figure 2.5(c)). Un exemple serait les télé-mesures.

2.4.1.3 Composite

Une communication composite comprend une communication point-à-multipoint et une communication multipoint-à-point. Dans ce type de communications, le trafic est bidirectionnel (voir figure 2.5(b)) mais les destinations ne communiquent pas entre elles. Un exemple est la consultation et mise à jour d'une base de données partagée.

2.4.1.4 Multipoint-à-multipoint

Dans une connexion multipoint-à-multipoint, tous les nœuds sont des émetteurs et des récepteurs de données (*full multipoint*) (voir figure 2.5(d)) et peuvent communiquer entre eux. La vidéo-conférence multi-parties en est un bon exemple.

2.4.2 Définitions de l'ATM Forum

Une connexion point-à-multipoint est une collection de VCs (Virtual Channel) et VP's (Virtual Path) avec des nœuds (commutateur) de terminaison associés à cette connexion [7]. Une telle connexion a les propriétés suivantes:

1. Un seul nœud joue le rôle de source. Si celui-ci envoie des données, des copies de ces données sont envoyées aux autres nœuds de la connexion, appelés « destinations ».
2. Il n'y a pas de voie de retour pour permettre aux nœuds destinataires d'envoyer des données à la source.
3. Les nœuds destinataires ne peuvent pas communiquer entre eux.

Une connexion point-à-multipoint est établie d'abord à l'aide d'une connexion point-à-point initiée par la source vers une des destinations. Après quoi, à l'initiative de la source, d'autres nœuds peuvent se joindre à la connexion moyennant des cellules de signalisation appropriées. Cette opération peut se faire simultanément pour plusieurs nœuds à la fois c.-à-d. qu'il n'est pas nécessaire qu'un nœud donné soit connecté pour passer au nœud suivant. D'autre part, à l'initiative de la source, un nœud destinataire peut quitter une connexion point-à-multipoint en tout temps toujours, à l'aide de cellules de signalisation appropriées. Toute la signalisation est contrôlée par la source.

Il faut noter cependant que les spécifications de l'ATM Forum [8] introduisent la possibilité que les destinations elles même puissent faire des requêtes pour joindre ou quitter un groupe multipoint.

Multipoint-à-multipoint Il y a deux façons d'établir des connexions multipoint-à-multipoint selon l'ATM Forum. Pour cela, chaque nœud voulant envoyer de l'information à un certain groupe de nœuds peut

1. établir une connexion point-à-multipoint avec tous les nœuds du groupe, tel que décrit ci-dessus. Avec une telle procédure, un groupe de N nœuds fera appel à N connexions point-à-multipoint ou
2. établir une connexion point-à-point avec un serveur. Ce dernier est la racine de la connexion point-à-multipoint du groupe avec lequel le nœud en question veut communiquer. Des copies de toutes les cellules envoyées à ce serveur sont retransmises à tous les membres du groupe. Avec une telle procédure, un groupe de N nœuds fera appel à N connexions point-à-point abonné-serveur et une connexion point-à-multipoint serveur-abonnés.

2.4.3 Entrelacement de cellules

Dans le cas où un arbre est partagé par tous les membres d'une communication multipoint, des connexions de type fusion (qui peuvent être vues comme des communications multipoint-à-point) présentent le problème d'entrelacement de cellules. Si des cellules en provenance de différentes sources sont multiplexées à un point donné (un commutateur) sur un même VC, il devient impossible de distinguer les cellules afin d'en reconnaître les sources. Ceci est illustré par la figure 2.6 où on a trois sources (A, B et C) dont les cellules sont transmises sur un VC commun. Remarquons que ce problème concerne aussi les communications multipoint-à-multipoint. En fait, il suffit d'avoir plus d'une source dans une connexion multipoint pour que le problème de distinction de cellules se pose.

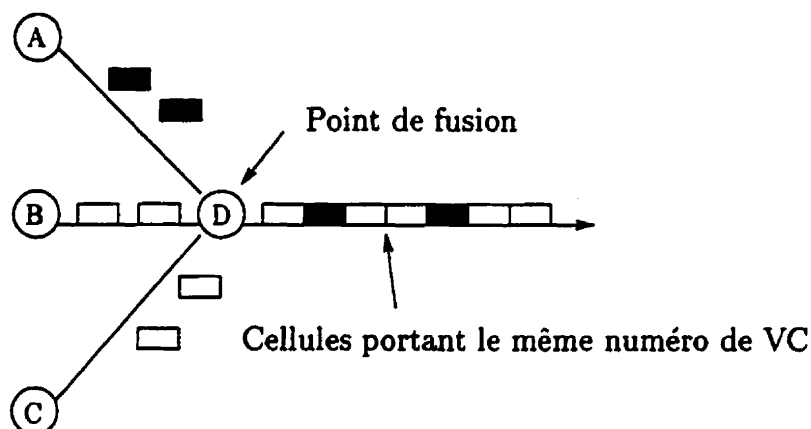


Figure 2.6: *Problème d'entrelacement de cellules.*

La difficulté de distinguer les cellules est causée d'une part, par le fait que le champ VPI/VCI qui permet d'identifier les cellules d'une connexion donnée est identique pour les cellules transmises sur le même VC et d'autre part, par le fait que l'AAL numéro 5 (ATM Adaptation Layer) [51], communément utilisée pour la transmission de données, ne permet pas d'identifier les sources des cellules et ne les numérote pas non plus. Afin de pallier ce problème, plusieurs solutions sont possibles présentant évidemment chacune des avantages et inconvénients. Parmi ces solutions, on distingue celles qui utilisent un arbre partagé et celles qui utilisent des arbres multiples.

2.4.3.1 Arbre partagé

On distingue trois solutions:

1. Utiliser les AAL numéro 3 et numéro 4 [51] qui ont un champ identificateur de multiplexage (MID: *Multiplexing Identifier*). Ce champ permet d'identifier les cellules de chaque source. Cependant, cette solution nécessite l'augmentation

du nombre de bits nécessaires à l'encapsulation, ce qui cause un gaspillage de largeur de bande et introduit un délai de traitement important [30].

2. Utiliser des jetons pour coordonner les sources. Ceci a été proposé dans [35] sous le nom de SMART (*Shared Many-to-many ReservaTion*). Dans cette solution, il y a un jeton par connexion. Ainsi, une seule source peut transmettre à la fois. Cette solution introduit un délai non négligeable notamment pour acquérir le jeton.
3. Utiliser des tampons au point de fusion de VC et faire attendre les cellules en provenance de sources différentes et n'en transmettre qu'un flot de cellules appartenant à une source à la fois. Cette solution est connue sous le nom de SEAM (*Scalable and Efficient ATM Multipoint-to-multipoint*) [43]. On a alors besoin d'un tampon par source. Cette solution semble être rapide [43] mais nécessite l'utilisation de mémoire supplémentaire et provoque des rafales de cellules très importantes avec des conséquences néfastes sur la qs.

2.4.3.2 Arbres multiples

On distingue deux solutions:

1. Utiliser un VC point-à-multipoint entre chaque source et ses destinations [23].
L'inconvénient de cette solution est qu'elle utilise un très grand nombre de numéros de VCI et peut ne pas être utilisable pour de grands réseaux.
2. Utilisation de serveurs où chaque source envoie ses cellules vers un serveur duplicateur sur un VC point-à-point [23]. Ce dernier se charge alors d'envoyer les cellules vers les destinations sur un VC point-à-multipoint. Cette solution

cause un problème de concentration de trafic autour du serveur qui peut engendrer une congestion importante.

2.4.3.3 Comparaison des deux approches

Une différence fondamentale entre l'approche avec arbre partagé et celle avec arbres multiples est que dans la première, si un terminal (source) se joint à une connexion en cours, le trafic écoulé sur un arc donné de l'arbre partagé peut augmenter. De plus, le trafic écoulé par un arc donné dépend de la topologie de l'arbre partagé. Par contre, la deuxième approche permet d'ajouter un terminal sans affecter le trafic écoulé par les arcs des différents arbres car le trafic écoulé par un arc donné ne dépend pas de la topologie de l'arbre qui l'utilise (car il y a une source par arbre) ce qui simplifie considérablement l'analyse du trafic écoulé par le réseau.

Ainsi, on a un problème de caractérisation du trafic écoulé par les arcs du réseau qui dépend de l'approche adoptée. Ce problème est très important et à notre connaissance, il n'a pas été étudié dans la littérature. C'est pourquoi nous allons y revenir en profondeur dans le chapitre 3 où nous allons décrire notre modèle.

2.4.4 Décomposition de connexions multipoint

Dans [52], on spécifie que la diffusion ou duplication de cellules ne peut se faire qu'entre des nœuds d'un même réseau. Ainsi, il n'est pas possible d'établir un VC point-à-multipoint contenant des nœuds appartenant à des réseaux distincts. On part bien entendu de l'hypothèse que le réseau est partitionné en sous-réseaux,

par exemple, des réseaux géographiquement éloignés. Les partitions sont connectées entre elles par des connexion point-à-point. L'objectif est surtout de simplifier l'étude des connexions multipoint en traitant chaque partition à part.

Pour qu'un nœud donné puisse diffuser des données à un groupe de nœuds situés dans un sous-réseaux différent, il doit d'abord établir une connexion point-à-point avec l'un des nœuds de ce sous-réseau qui ensuite peut diffuser les cellules aux autres nœuds du groupe. (Cela nous rappelle l'idée de configuration en îles de Mbone.) Ainsi, une connexion multipoint impliquant des nœuds de différents réseaux est décomposée en plusieurs niveaux. On parle alors de *décomposition* de connexions multipoint [52]. Ceci est illustré par la figure 2.7 où le nœud A du réseau 1 établit une connexion point-à-point avec le nœud C (via le nœud B) qui diffuse les cellules aux nœuds D et E du réseau 2. Nous avons alors décomposé la connexion en une connexion point-à-point de A vers C et en une connexion point-à-multipoint de C vers D et E.

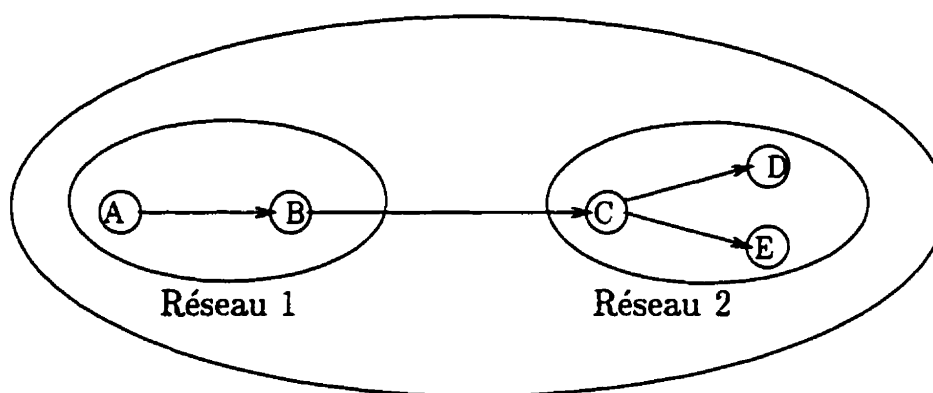


Figure 2.7: *Décomposition de connexions multipoint en ATM.*

2.4.5 Qualité de service

Dans un réseau de type ATM, on retrouve la notion de qualité de service (QS) du niveau appel et celle du niveau cellule (qs). La qs est souvent liée à la perte de cellules et au délai de bout en bout. Cette qs est déclarée par l'utilisateur. La perte d'appels (QS) dépend alors de la qs. En effet, en ATM on utilise des procédures d'acceptation d'appels (CAC) afin de savoir si un appel donné peut être accepté, tout en respectant la qs du niveau cellule et du nouvel appel et des connexions en cours.

Afin d'être en mesure d'évaluer la perte d'appels (souvent en considérant les probabilités stationnaires de perte d'appel), on a besoin de connaître les probabilités de blocage sur les artères de transmission. On a alors introduit la notion de débit équivalent [59, 66, 27] dans le contexte du point-à-point. Cette notion permet de simplifier considérablement la synthèse de réseaux avec circuit virtuel dans la mesure où cela nous permet d'utiliser les résultats obtenus pour les réseaux multi-débit à commutation de circuits.

Pour un arc de capacité C et pour un type d'appel donné, une définition simplifiée du débit équivalent est $W = C/n$ où n est le plus grand nombre d'appels pouvant être écoulés sur l'arc en question, tout en respectant les contraintes de qs. Ainsi, nous pouvons évaluer les probabilités de blocage sur les arcs.

Il n'y a pas encore une définition claire de la qs d'une connexion multipoint. Selon l'ATM Forum [7], la qs déclarée par le nœud source sera celle des destinations. En d'autres mots, les destinations ne peuvent imposer des contraintes de qs plus ou moins exigeantes que celle de la source. De plus, ces nœuds destinataires doivent, selon l'ATM Forum, avoir les mêmes descripteurs de trafic que la source (CBR,

VBR ou ABR) ainsi que les mêmes paramètres, comme par exemple le débit dans une connexion CBR et le débit moyen dans une connexion VBR. Ainsi, dans une communication multipoint, tous les terminaux doivent être en mesure de supporter le débit et la qs imposés par la source.

2.4.6 Adressage

Il n'y a malheureusement aucune spécification de l'adressage de groupes en ATM. Cela est sûrement dû à la difficulté de trouver un emplacement dans l'en-tête des cellules pour un adressage multipoint convenable dans la mesure où l'espace réservé aux VPI/VCI ne prévoit pas l'utilisation d'un identificateur de groupe. En effet, dans une connexion ATM, on utilise une succession de VP et de VC (si on a au moins un nœud de transit) et les VPI et les VCI peuvent changer en traversant ces nœuds de transit rendant très difficile l'utilisation d'un numéro unique pour identifier une connexion à plusieurs participants.

D'autre part, par opposition avec Internet, dans les réseaux de type ATM, une source doit connaître tous les membres d'un groupe auxquels les cellules seront envoyées. C'est donc la source qui contrôle l'appartenance d'un membre donné au groupe. On peut voir ici le caractère connecté [3] (*connection oriented*) du multipoint les réseaux ATM dans la mesure où on a une connexion de bout-en-bout entre la source et les destinations. Cela cause un problème d'incompatibilité important lorsqu'on veut véhiculer du trafic multipoint d'Internet sur un réseau de type ATM.

2.4.7 Résumé

Nous pouvons constater que les réseaux de type ATM offrent deux avantages principaux: 1) la grande disponibilité de largeur de bande et 2) la garantie de qs. Ces deux avantages sont nécessaires pour réaliser des communications multipoint où le trafic du niveau cellules consomme beaucoup de largeur de bande et où il est nécessaire d'offrir une qs garantie, chose qu'Internet ne peut offrir aujourd'hui. Malheureusement, les infrastructures déjà en place (Internet) ne permettent pas de passer facilement d'Internet à la technique ATM et le principal obstacle est le coût très élevé qui résulterait d'un tel passage. C'est pourquoi, on a pensé à utiliser la technique ATM comme véhicule pour le trafic multipoint d'Internet plutôt que remplacer Internet par ATM.

2.5 Trafic Internet multipoint sur ATM

Dans la littérature, il y a un nombre considérable de travaux effectués pour trouver comment véhiculer le trafic multipoint d'Internet par des réseaux de type ATM [5, 13, 6, 4]. Cela présente beaucoup d'avantages notamment, de bénéficier des infrastructures et des applications déjà en place d'Internet et profiter de la qualité de service et la largeur de bande offertes par les réseaux de type ATM. Malheureusement, les spécifications relatives au multipoint pour les réseaux de type ATM et celles pour Internet ne sont pas compatibles.

D'une part, les communications point-à-multipoint dans les réseaux de type ATM sont contrôlées par la source et le scénario multipoint-à-multipoint nécessite autant d'arbres qu'il y a de sources. D'autre part, les communications point-à-multipoint

dans Internet sont contrôlées par les destinations et le scénario multipoint-à-multipoint fait appel à un seul arbre en utilisant les serveurs dupicateurs. Ces différences causent des problèmes importants d'adressage et de signalisation. Dans cette section, nous décrivons brièvement les problèmes essentiels liés à l'utilisation de l'infrastructure multipoint ATM pour véhiculer le trafic multipoint d'Internet.

2.5.1 Adressage

Le problème le plus important rencontré pour acheminer du trafic multipoint venant d'Internet sur un réseau de type ATM (si on veut respecter les spécifications de l'ATM Forum) est l'adressage. Les caractères connecté des réseaux ATM et non connecté d'Internet en sont la cause principale.

Dans les spécifications de l'ATM Forum, les connexions point-à-multipoint sont gérées par la source et c'est cette source qui garde le contrôle de la connexion en ce qui concerne les membres qui participent à la connexion. De l'autre côté, dans le réseau Internet, ce sont les destinations qui sont responsables de leur appartenance aux groupes multipoint et la source peut ne pas savoir qui recevra les données qu'elle envoie. Ainsi, il faut introduire un mécanisme sophistiqué afin d'être en mesure d'acheminer le trafic multipoint provenant d'Internet en utilisant l'infrastructure multipoint de l'ATM.

Pour résoudre le problème d'adressage, l'IETF (*Internet Engineering Task Force*) a introduit la notion de serveur MARS (*Multicast Address Resolution Server*) [5, 13]. Un serveur MARS contient un registre où sont sauvegardées les différentes informations concernant les différents groupes multipoint et leurs membres. Ces informations incluent les adresses ATM de tous les nœuds faisant partie d'un groupe

multipoint Internet donné ainsi que l'adresse IP du groupe multipoint. Les serveurs MARS sont chargés d'informer chaque membre d'un groupe multipoint donné de l'identité des autres membres faisant partie de ce groupe. Un nœud donné interroge un serveur MARS lorsqu'il désire connaître les adresses ATM des membres d'un groupe multipoint donné. Lorsqu'un nœud donné désire joindre ou quitter un groupe multipoint, il doit informer un serveur MARS. Ce dernier se charge alors d'informer tous les membres de la connexion multipoint de la nouvelle composition du groupe participant.

2.5.2 Comment réaliser le multipoint-à-multipoint?

Deux solutions sont retenues pour permettre au trafic multipoint-à-multipoint d'être véhiculé par le réseau: une première basée sur les serveurs dupicateurs et une autre sur un ensemble de VC point-à-multipoint.

2.5.2.1 Serveurs dupicateurs

Dans cette solution, chaque membre d'un groupe multipoint établit un VC point-à-point avec un serveur dupicateur. Chaque serveur dupicateur établit une connexion point-à-point avec un serveur MARS. Ce dernier ajoute alors le serveur à un VC point-à-multipoint unidirectionnel, où seuls les serveurs dupicateurs sont les destinations, sur lequel sont envoyées les informations relatives aux membres participant dans les différents groupes multipoint. Un serveur dupicateur donné se charge d'envoyer des copies des cellules reçues aux membres d'un même groupe sur un VC point-à-multipoint et aux autres serveurs sur des VC point-à-point. Ceci est illustré dans la figure 2.8. Cette solution a les avantages suivants:

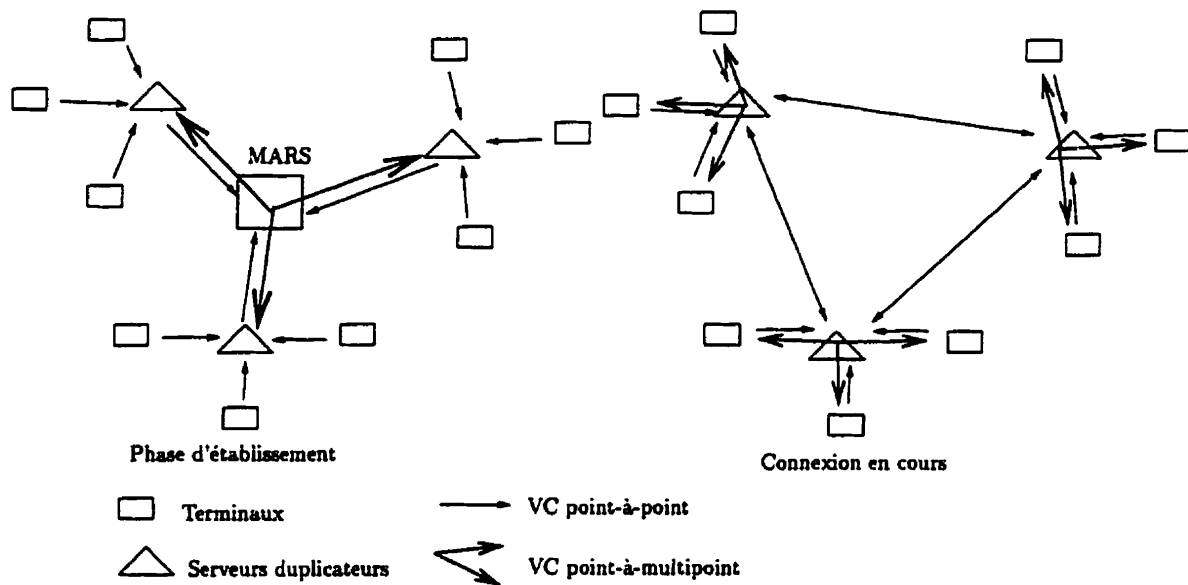


Figure 2.8: *Solution avec serveurs dupicateurs.*

- économie de VCI (identificateurs de VC) et
- peu de trafic de signalisation en cas de changement dans les membres de groupes multipoint.

Cette méthode a les inconvénients suivants:

- surcharge des serveurs dupicateurs et concentration de la congestion aux alentours de ces serveurs,
- les arbres générés pour connecter les nœuds (commutateurs) ne sont pas nécessairement optimaux en termes d'utilisation de ressources,
- introduction d'un délai important dû au passage obligatoire par les serveurs et

- possibilité de renvoi d'une copie d'un message à celui qui l'a envoyé. Une solution serait alors d'établir un VC point-à-point entre le serveur et chaque destination mais cela gaspille les champs VCI.

2.5.2.2 Ensemble de VCs

Dans cette solution connue sous le nom de *VC mesh* [23] en anglais, on propose de générer un VC point-à-multipoint pour chaque source utilisant ainsi un arbre par source. La duplication de cellules se fait alors dans les commutateurs ATM. Chaque source dans un sous-réseau appartenant à Internet qui veut envoyer des données à un groupe de nœuds doit faire une requête auprès du serveur MARS afin d'obtenir les adresses ATM des nœuds à connecter et établir un VC point-à-multipoint avec ces nœuds. Ceci est illustré par la figure 2.9 où le nœud *S* fait une requête après du MARS pour joindre un groupe multipoint. Cette solution est préférable à celle avec serveurs dupicateurs car

- elle est plus facile à utiliser avec la signalisation proposée par l'ATM Forum,
- elle offre une meilleure qualité de service car on a un arbre pour chaque source de données et
- il n'est pas possible qu'un membre donné reçoive ses propres messages.

Les inconvénients de cette solution sont

- le gaspillage de numéros de VC et
- le gaspillage de largeur de bande si les VCs point-à-multipoint sont sous utilisés surtout dans des communications multipoint-à-multipoint où peu de sources sont actives à la fois.

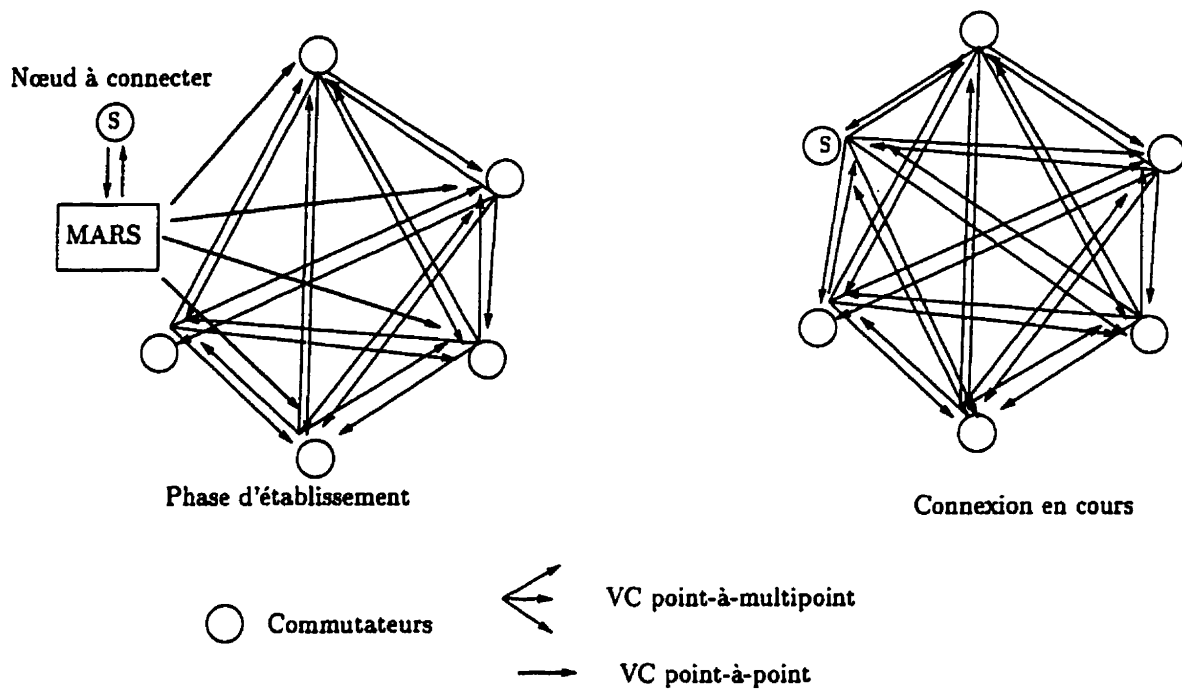


Figure 2.9: *Solution avec un ensemble de VC.*

2.5.3 Résumé

Utiliser l'infrastructure ATM pour véhiculer du trafic multipoint d'Internet est faisable. Malheureusement, à cause de l'incompatibilité de la signalisation et de l'adressage, il a fallu introduire des mécanismes sophistiqués pour pallier cette incompatibilité. Soulignons par ailleurs que l'ATM Forum et l'IETF sont entrain de définir de nouvelles recommandations qui permettent de réduire les incompatibilités et d'introduire plus de souplesse afin de pouvoir transporter des datagrammes IP sur un réseau de type ATM [8, 23]. Par exemple, on sait que l'ATM Forum a donné la possibilité aux destinations de gérer leur appartenance aux groupes multipoint.

2.6 Conclusion

Dans ce chapitre, nous avons donné une description du multipoint en général et nous nous sommes particulièrement attardés sur le réseau Internet et les réseaux de type ATM. Nous avons donné une brève description des problèmes que l'on rencontre pour réaliser les communications multipoint. Nous avons sélectionné quelques solutions intéressantes que certains chercheurs proposent afin de surmonter certains de ces problèmes. Un des problèmes est celui de la synthèse de réseau, problème qui jusqu'à maintenant n'a pas été abordé dans la littérature. Cela constitue l'objet de notre travail de recherche. Le problème de synthèse inclut plusieurs problèmes notamment l'acheminement et le dimensionnement. Dans les trois chapitres suivants, nous donnons une description de ces problèmes ainsi que de la démarche que nous proposons pour les résoudre.

CHAPITRE 3

Modélisation du problème de synthèse de réseaux

Dans ce chapitre, nous décrivons le modèle que nous avons utilisé pour étudier le problème de synthèse de réseaux en justifiant nos choix et hypothèses. Dans la section 3.1, nous décrivons différents scénarios multipoint dont nous dégageons le scénario le plus pertinent qu'on se propose d'étudier. Dans la section 3.2, nous proposons une formulation mathématique du modèle d'acheminement dans le cas où les arbres utilisés pour connecter les appels sont donnés. Cela nous permettra de comprendre le problème d'acheminement dans un premier temps et ensuite de généraliser le modèle au cas où les arbres ne sont pas connus d'avance. Dans la section 3.3, nous décrivons le modèle utilisé pour le problème de synthèse, aussi dans le cas où les arbres sont connus. Enfin, dans la section 3.4, nous décrivons les algorithmes que nous proposons pour résoudre les problèmes d'acheminement d'appels et de synthèse de réseaux, pour un ensemble d'arbres donné.

3.1 Problèmes multipoint pertinents

Dans la mesure où il existe une grande diversité d'applications qui font appel au multipoint (diffusion multimédia, vidéo-conférence, jeux interactifs, etc), il est difficile de représenter toutes ces applications par un seul modèle. Nous voulons savoir quel est le problème multipoint qui serait le plus intéressant à considérer en premier lieu. Notre critère se base essentiellement sur l'impact de l'introduction des communications multipoint sur les performances du réseau commuté.

3.1.1 Classification des communications multipoint

Nous définissons quatre attributs qui permettent de caractériser les communications multipoint:

1. le *type*: on distingue le type point-à-multipoint, le type multipoint-à-point, le type composite et le type multipoint-à-multipoint,
2. le *dynamisme* des connexions: on distingue les connexions statiques et les connexions dynamiques c.-à-d., où les usagers peuvent joindre ou quitter une connexion multipoint pendant la durée de vie de celle-ci,
3. la *synchronisation* permet de savoir si une application donnée est en temps réel ou pas auquel cas il faudra synchroniser les usagers ou pas et
4. la *politique* (en cas de plusieurs sources) selon laquelle les usagers peuvent transmettre simultanément ou uniquement un à la fois.

Nous allons illustrer comment, selon l'application considérée, certains attributs peuvent être ou ne pas être pertinents quant aux ressources utilisées dans le réseau commuté pour établir des connexions multipoint.

3.1.2 Point-à-multipoint

Considérons une diffusion de données telle que la diffusion de cours, de conférences, de films, etc. Le sens des données est unique, de la source vers les destinations et aucune synchronisation ni politique de transmission ne sont requises. La variation de l'ensemble d'utilisateurs a une influence négligeable sur la gestion de ressources. En effet, supposons que nous connaissons les commutateurs à travers desquels un certain nombre d'utilisateurs reçoivent les données diffusées par une source donnée. Si de nouveaux utilisateurs rejoignent ou quittent la connexion à travers les mêmes commutateurs déjà utilisés, cela n'affecte pas le trafic écoulé par l'arbre qui connecte l'ensemble de ces commutateurs dans la mesure où peu importe le nombre de terminaux connectés aux commutateurs, une seule copie d'une unité de données sera envoyée aux commutateurs. Des exemples typiques sont la diffusion de cours à des sites connus et la diffusion de conférences où on diffuse les données peu importe le nombre d'utilisateurs connectés aux commutateurs, pourvu qu'il y ait au moins un utilisateur par commutateur. Ceci est illustré par la figure 3.1 où le fait que l'ensemble de destinations change de $\{A, C, D, E, F\}$ à $\{B, C, D, E, F, G\}$ n'affecte pas le trafic écoulé dans le réseau commuté (liens dessinés en lignes épaisses). Dans chaque cas, seule une copie d'une unité de données en provenance de la source S sera écoulee sur les liens du réseau commuté (représentés en lignes épaisses) vers les destinations.

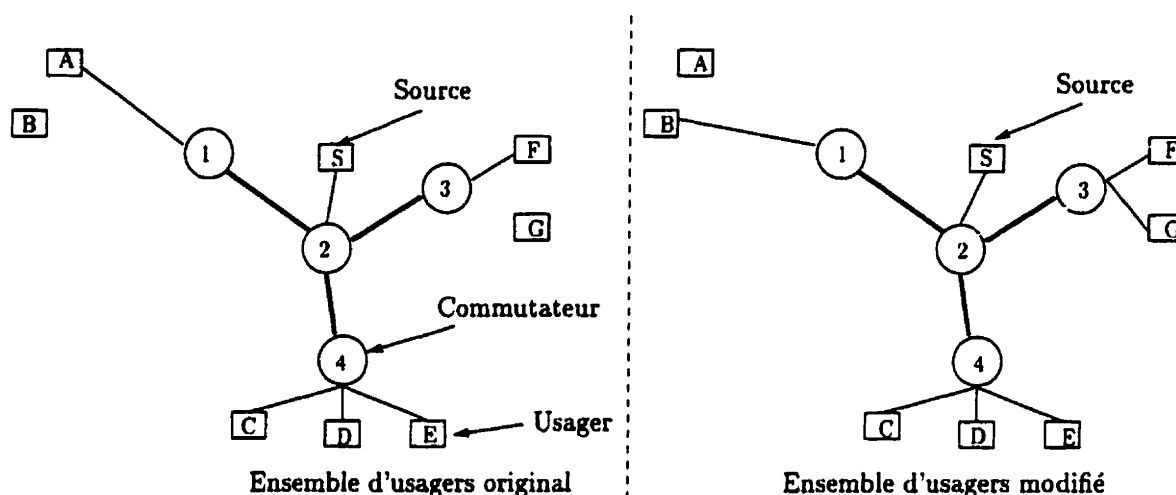


Figure 3.1: *Le trafic dans le réseau commuté ne dépend pas du changement des membres.*

3.1.3 Multipoint-à-multipoint

Considérons une application du type vidéo conférence. Dans ce type d'applications, il est fort probable que la politique de transmission ne permette qu'à un usager à la fois de transmettre. En effet, comme il est de coutume dans les discussions où les intervenants sont des êtres humains, une personne parle et les autres écoutent. De plus, la direction de transmission peut changer au cours du temps car nous avons plusieurs sources.

D'autre part, la synchronisation est souhaitée dans ce type d'application. Afin d'établir une connexion multipoint pour ce type d'application, la solution avec arbre partagé est appropriée. On utilise alors un jeton pour la politique de transmission des usagers tel que décrit dans la section 2.4.3.1.

Par ailleurs, comme les membres des groupes multipoint peuvent changer, le dynamisme des connexions est pertinent dans ce cas pour deux raisons. La première

du nombre d'utilisateurs connectés au nœud 3. On a alors un problème de caractérisation du débit équivalent (voir section 2.3.5) qui dépend du nombre d'utilisateurs connectés à un instant donné et de la topologie de l'arbre qui connecte ces utilisateurs.

Le problème de caractérisation du débit équivalent dépend aussi de la politique de transmission. Supposons qu'on ait un débit équivalent W sur un arc donné, pour la connexion multipoint donnée dans la figure 3.2. Par exemple, si on veut que ce débit équivalent ne soit pas dépendant du nombre d'utilisateurs qui transmettent au nœud 3, on peut ajuster le débit x alloué à chaque utilisateur (en supposant que les utilisateurs transmettent au même débit, on peut ajuster la vitesse de rotation du jeton) de sorte qu'on ait, en tout temps, $W = ix$, où i est le nombre d'utilisateurs connectés au nœud 3. Par contre, si on veut allouer à chaque utilisateur un débit y indépendant du nombre d'utilisateurs qui transmettent à partir du nœud 3, le débit $Y = iy$ va augmenter avec le nombre d'utilisateurs rendant la notion de débit équivalent très difficile à utiliser voire, inappropriée dans ce cas. Nous avons alors besoin d'introduire de nouveaux modèles pour caractériser le débit équivalent dans ce contexte particulier.

3.1.4 Multipoint-à-point

On considère des applications du type fusion comme les télé-mesures. On peut rencontrer dans ce genre de communications les mêmes problèmes connus pour le type multipoint-à-multipoint. Cependant, il est important de constater que dans la mesure où on a une seule destination, aucune duplication d'unités de données n'est requise et on peut alors réaliser ce type de communication par de simples connexions point-à-point.

Il est difficile de savoir dans quelle mesure il serait avantageux de réaliser le

multipoint-à-point à l'aide de plusieurs connexions point-à-point indépendantes car cela dépend des applications. Par exemple, si une collecte de données doit être effectuée dans un certain ordre, faire des connexions point-à-point indépendantes n'est pas la meilleure solution. Il faudrait plutôt utiliser des mécanismes à jeton pour synchroniser les sources ce qui introduit une dépendance entre les sources et par conséquent, une corrélation dans le trafic qu'il faudra éventuellement modéliser.

3.1.5 Temps réel

Si on considère une application telle qu'un jeu multimédia multi-usagers distribué (simulation de batailles de tanks), les communications sont en général du type multipoint-à-multipoint, les connexions sont dynamiques (par exemple, si un joueur perd, il quitte le jeu), la synchronisation est primordiale et la politique de transmission doit permettre aux usagers de transmettre des données simultanément. Ce type d'applications est le plus exigeant en termes de ressources. Pour que toutes ces exigences soient satisfaites, on doit faire appel à un ensemble de connexions point-à-multipoint c.-à-d., chaque joueur établit une connexion point-à-multipoint avec tous les autres joueurs.

Heureusement, nous ne connaissons pas d'exemple où ce type de communication se fait sur une grande étendue géographique et ainsi, on peut s'attendre à ce que le nombre de nœuds (commutateurs) utilisés soit très petit voir nul (si ça se passe dans un réseau local) faisant en sorte que ce type d'applications n'affecte pas considérablement la synthèse du réseau commuté. Néanmoins, dans la mesure où la technologie des réseaux continue à évoluer notamment, la disponibilité de la largeur de bande, de telles applications deviendront envisageables à grande échelle. Cela

nécessiterait une modélisation adéquate notamment, comment faire un ensemble de connexions point-à-multipoint synchronisées entre elles.

3.1.6 Résumé

D'après cette brève discussion, nous pouvons conclure que les applications de type diffusion (point-à-multipoint) sont très utilisées actuellement. Elles sont aussi simples à modéliser dans la mesure où on n'a pas vraiment à considérer l'aspect dynamique des connexions et où on peut appliquer la notion de débit équivalent sans difficulté particulière. De plus, le type multipoint-à-multipoint (plusieurs sources) peut être réalisé à l'aide d'un ensemble de connexions point-à-multipoint donnant à ce dernier encore plus d'intérêt. C'est pourquoi, nous avons choisi de nous y intéresser ici. Dans le cas où on a plusieurs sources et qu'on utilise un arbre partagé, le problème de synthèse devient très compliqué à cause notamment de la difficulté de gérer des connexions dynamiques d'une part et la difficulté de caractériser le débit équivalent, d'autre part. Ces sujets devront faire l'objet d'études plus poussées qui sortent du cadre de cette thèse.

3.2 Description du modèle

Le modèle que nous décrivons dans cette section a été introduit dans [40]. Nous supposons que nous avons un réseau à commutation de circuits. Le choix de la commutation de circuits ne limite pas le modèle aux réseaux de téléphone public mais le modèle peut être appliqué pour les réseaux de type ATM en utilisant la notion de débit équivalent [59].

Dans la mesure où on peut avoir des connexions multipoint qui requièrent des

largeurs de bande différentes, nous supposons que nous avons C classes d'appels, chaque classe représentant un débit différent. Pour chaque classe d'indice c , nous définissons $W(c)$ comme étant le nombre de serveurs saisis et libérés simultanément au début et en fin de connexion, respectivement.

Nous supposons que nous avons K types d'appels par classe, où un type correspond à un ensemble de nœuds constitué d'une source et d'un nombre arbitraire, mais fixe, de destinations, équivalent à un couple origine-destination en point-à-point. Nous introduisons k comme étant l'indice de type d'appel.

Nous supposons donc que le modèle traite uniquement le type point-à-multipoint étant donné l'intérêt qu'il présente. Il est alors possible de connaître à priori les commutateurs qui doivent être connectés pour chaque type d'appel et ce, en considérant les usagers potentiels qui sont raccordés au réseau par l'intermédiaire de ces commutateurs.

De plus, nous n'avons pas à modéliser l'aspect dynamique des connexions. En effet, dans la mesure où nous considérons des applications de type diffusion et que tous les commutateurs peuvent dupliquer des cellules, nous supposons que le réseau commuté n'est pas affecté par la modification des ensembles d'usagers participant aux communications multipoint (voir section 3.1).

Nous supposons aussi que nous connaissons la *matrice de trafic* généralisée $A^k(c)$ qui représente le rapport du taux d'arrivée poissonnien des appels de type k et de classe c , au taux de service exponentiel. $A^k(c)$ est aussi appelée *demande* de trafic. Nous supposons que les matrices de trafic sont déterminées par une prévision de la demande, ce qui est possible pour des applications comme la diffusion de conférences, de cours à distance et de films (voir section 2.2.1). En pratique, la prévision et la

représentation de ces matrices de trafic présente un problème non trivial surtout à cause de leur nombre potentiellement élevé mais nous allons supposer que nous les connaissons afin d'être en mesure d'étudier l'impact du multipoint sur la synthèse du réseau.

3.2.1 Acheminement

Nous supposons que nous utilisons un acheminement simple par partage de charge en étant conscients que cet acheminement n'est pas très utilisé en pratique. En effet, en pratique on utilise plutôt des techniques d'acheminement adaptatif [38] qui tiennent compte de l'état du réseau, tel que la technique DCR (*Dynamically Controlled Routing*). Malheureusement, il est très difficile de faire une étude mathématique rigoureuse en utilisant des techniques d'acheminement adaptatif et c'est pour quoi nous nous contentons d'un acheminement par partage de charge simple. Ceci va nous permettre de bien comprendre le problème de synthèse de réseau et par la suite, le problème de sélection d'arbres.

Nous supposons qu'à chaque couple (k, c) correspond un ensemble $\mathcal{S}^k(c)$ connu et fixe d'arbres ayant une racine au nœud source, que nous dénoterons (k, c, m) où l'indice m dénote le numéro de l'arbre dans cet ensemble. Un arbre (k, c, m) sert à connecter les appels de type (k, c) . On a ainsi plusieurs choix d'arbres pour connecter un appel donné. Pour simplifier le modèle, nous supposons que nous avons M arbres pour chaque couple (k, c) . Nous introduisons aussi $\mathcal{S} = \bigcup_{k,c} \mathcal{S}^k(c)$.

Nous définissons les *coefficients de partage de charge* $\alpha_m^k(c)$ qui représentent la fraction des appels multipoint de type k et de classe c qui sont offerts à l'arbre (k, c, m) . En pratique, lorsqu'un appel se présente, on choisira l'arbre auquel on va

l'offrir par un tirage aléatoire selon les probabilités $\alpha_m^k(c)$. On tente de connecter l'appel sur l'arbre ainsi choisi et si l'appel est bloqué sur cet arbre, alors il est perdu.

3.2.2 Calcul du blocage

Un élément important dans l'optimisation de l'acheminement est l'évaluation de la qualité de service (QS) du niveau appel pour un réseau donné. Cette QS s'exprime comme la probabilité de perte pour les appels de type k et de classe c , dénotée $L^k(c)$. Nous supposons que la qualité de service (qs) du niveau paquet dans le cas de réseaux ATM est contenue dans le débit équivalent associé à une classe d'appel c .

Il faut d'abord calculer la probabilité de perte d'un appel sur un arbre donné, ce qui constitue la mesure de la QS. Nous faisons l'hypothèse suivante:

un appel ne sera accepté que si on peut le connecter sur tous les arcs de l'arbre.

Cette hypothèse signifie que les nœuds participants doivent tous être connectés sinon, l'appel est perdu. Nous faisons cette hypothèse pour simplifier le modèle et il n'est pas difficile de faire en sorte que l'on puisse accepter des connexions partielles où on ne connecte pas nécessairement tous les nœuds.

Nous définissons $L_m^k(c)$ comme étant la probabilité de perte sur l'arbre (k, c, m) . Nous pouvons écrire

$$L_m^k(c) = 1 - Q_m^k(c) \quad (3.1)$$

$$Q_m^k(c) = \prod_{s \in (k, c, m)} [1 - B_s(c)] \quad (3.2)$$

où $B_s(c)$ est la probabilité de blocage d'un appel de classe c sur l'arc s et $Q_m^k(c)$ est

la probabilité de connexion sur l'arbre (k, c, m) . Remarquons aussi que l'équation (3.2) repose sur l'hypothèse d'indépendance:

les événements de blocage sur les arcs du réseau sont indépendants.

Nous faisons cette hypothèse pour simplifier le problème. En effet, en pratique les probabilités de blocage sur les arcs du réseau sont sûrement dépendantes même pour le point-à-point. Cette dépendance pourrait être accentuée par l'aspect multipoint à cause de la corrélation du trafic du niveau cellules. Notons cependant qu'aucune étude n'a montré la validité ou la non validité d'une telle hypothèse. Pour le point-à-point, le fait que les résultats obtenus sous cette hypothèse sont confirmés par les mesures dans le réseau justifie son utilisation mais nous ne savons pas si cela sera le cas pour le multipoint aussi. Nous la maintenons en étant conscients que tôt ou tard, il faudra la remettre en question.

On peut écrire, pour des coefficients de partage de charge $\alpha_m^k(c)$ donnés,

$$L^k(c) = \sum_m \alpha_m^k(c) L_m^k(c). \quad (3.3)$$

Le calcul des probabilités de perte se ramène donc au calcul des probabilités de blocage de toutes les classes $B_s(c)$ sur les arcs du réseau. Nous posons $a_s(c)$ le trafic total de classe c (mesuré en nombre moyen de demandes de connexion) offert à l'arc s et $\bar{a}_s(c)$ le trafic total écoulé sur cet arc. Pour l'arc s et la classe c , nous pouvons alors écrire

$$\begin{aligned} \bar{a}_s(c) &= \sum_{k,m} A^k(c) \alpha_m^k(c) Q_m^k(c) \mathcal{I}_{s,(k,c,m)} \\ \mathcal{I}_{s,(k,c,m)} &= \begin{cases} 1 & \text{si l'arc } s \text{ est dans l'arbre } (k, c, m) \\ 0 & \text{sinon} \end{cases} \end{aligned} \quad (3.4)$$

$$a_s(c) = \frac{\bar{a}_s(c)}{1 - B_s(c)} \quad (3.5)$$

où la matrice \mathcal{I} est la matrice d'incidence arc-arbre.

Pour simplifier le problème, nous avons supposé que les arcs sont symétriques c.-à-d., le trafic $a_s(c)$ sur l'arc s connectant la paire de nœuds (i, j) désigne la somme des trafics écoulés de i vers j et de j vers i . En pratique, les arcs dans un réseau de télécommunications peuvent être asymétriques (comme c'est le cas des VCs en ATM) donnant lieu à un graphe orienté, ce qui complique considérablement le problème de génération d'arbres [85].

Nous supposons que nous connaissons une fonction g qui nous permet de calculer la probabilité de blocage sur les arcs en fonction des trafics offerts, des débits et des autres paramètres de l'arc donnée par

$$B_s(c) = g_s^c(\mathbf{a}_s, \mathbf{W}, N) \quad (3.6)$$

où \mathbf{a}_s dénote le vecteur des trafics offerts à l'arc s et \mathbf{W} le vecteur des débits des différentes classes d'appels.

Dans [56], on propose un algorithme numérique complexe pour calculer les $B_s(c)$ de façon exacte. Ici, nous utilisons l'approximation donnée dans [64] pour évaluer $B_s(c)$. Dans le cas d'une seule classe d'appel, $B_s(c)$ est donnée par la formule d'Erlang.

Il est évident que nous devons résoudre le système d'équations (3.2–3.6) pour obtenir les valeurs des probabilités de blocage. Nous avons donc un système de point fixe. On sait que dans le contexte du point-à-point et dans le cas de plusieurs classes d'appel, ce système peut posséder plusieurs solutions qui se manifestent par une hystérésis lors du fonctionnement du réseau [39] ce qui est susceptible d'être

vrai dans le contexte du multipoint aussi. Cependant, on peut facilement montrer que le théorème de Kelly [57] sur l'unicité des solutions du point fixe d'Erlang dans le cas du partage de charge est valable pour les communications multipoint telles que définies ci-dessus dans le cas où on a une seule classe d'appels. Nous ne savons pas si c'est vrai dans le cas où on a plus d'une classe d'appel.

La résolution du point fixe se fait comme suit: soit $\mathbf{B}^{(i)}$ et $\mathbf{a}^{(i)}$ respectivement le vecteur de probabilités de blocages et le vecteur de trafics offerts aux arcs à l'itération i de l'algorithme de résolution du point fixe. En partant du vecteur de probabilités de blocage initial $\mathbf{B}^{(0)}$ pour $i = 0$,

1. calculer $\mathbf{a}^{(i+1)}$ à l'aide de l'équation (3.5),
2. calculer $\mathbf{B}^{(i+1)}$ à l'aide de l'équation (3.6),
3. si $\max_s |a_s^{(i+1)} - a_s^{(i)}| < \varepsilon_1$ et $\max_s |B_s^{(i+1)} - B_s^{(i)}| < \varepsilon_2$: fin, sinon $i = i + 1$,
retour à 1.

où $\varepsilon_1 \ll 1$ et $\varepsilon_2 \ll 1$ donnent la précision sur la résolution du point fixe.

3.2.3 Optimisation de l'acheminement

L'optimisation de l'acheminement est pertinente parce qu'en pratique, les valeurs courantes des demandes de trafic sont différentes des prévisions de trafic utilisées pour dimensionner le réseau et varient sur une échelle de temps assez courte, typiquement, une heure. Puisqu'on ne peut changer aussi fréquemment les valeurs des capacités des liens, la seule façon d'améliorer les performances du réseau est d'optimiser l'acheminement c.-à-d., recalculer les coefficients de partage de charge.

Notons qu'il ne s'agit pas d'un acheminement adaptatif où les coefficients de partage de charge sont modifiés très fréquemment en fonction de l'état du réseau mais il s'agit plutôt d'un acheminement statique sur une période de quelques heures.

Nous pouvons maintenant écrire le problème de l'optimisation de l'acheminement statique en supposant que nous voulons maximiser le revenu total produit par le réseau. Nous avons alors le problème

$$\min_{\alpha} z = - \sum_{k,c} w^k(c) \bar{A}^k(c) \quad (3.7)$$

$$\sum_m \alpha_m^k(c) = 1 \quad (3.8)$$

$$\alpha_m^k(c) \geq 0 \quad (3.9)$$

où $w^k(c)$ représente le revenu produit par un appel de type k et de classe c écoulé dans le réseau et $\bar{A}^k(c)$ est le trafic de classe c et de type k écoulé dans le réseau. Sa valeur est donnée par

$$\bar{A}^k(c) = A^k(c) \sum_m \alpha_m^k(c) Q_m^k(c). \quad (3.10)$$

Nous pouvons incorporer les équations du point fixe comme contraintes et relaxer les variables $B_s(c)$ comme variables indépendantes pour obtenir le problème équivalent

$$\min_{\alpha, B} z = - \sum_{k,c} w^k(c) \bar{A}^k(c) \quad (3.11)$$

$$\sum_m \alpha_m^k(c) = 1 \quad (v^k(c)) \quad (3.12)$$

$$\alpha_m^k(c) \geq 0 \quad (u_m^k(c)) \quad (3.13)$$

$$B_s(c) = g_s^c(a_s, W, N_s) \quad (y_s(c)) \quad (3.14)$$

où les quantités entre parenthèses sont les multiplicateurs de Kuhn-Tucker correspondant aux contraintes.

Nous allons maintenant écrire les équations d'optimalité pour les deux groupes de variables α et B . Le Lagrangien du problème est donné par

$$\begin{aligned} \mathcal{L}(\alpha, B, v, u, y) = & - \sum_{k,c} w^k(c) \bar{A}^k(c) + \sum_{k,c} v^k(c) \left[\sum_m \alpha_m^k(c) - 1 \right] \\ & - \sum_{k,c,m} u_m^k(c) \alpha_m^k(c) + \sum_{s,c} y_s(c) [B_s(c) - g_s^c(a_s, W, N_s)]. \end{aligned}$$

Après un long développement [40], on peut montrer que pour tous les arbres m du type k et de la classe c , la condition d'optimalité obtenue pour les variables α s'écrit

$$\begin{aligned} v^k(c) = & u_m^k(c) + A^k(c) Q_m^k(c) \left[w^k(c) + \right. \\ & \left. \sum_s \frac{1}{[1 - B_s(c)]} \sum_b y_s(b) \frac{\partial g_s^b}{\partial a_s(c)} \mathcal{I}_{s,(k,c,m)} \right] \quad \forall m. \end{aligned} \quad (3.15)$$

On peut donc identifier le coût induit $r_s(c)$ d'un arc s pour la classe c comme

$$r_s(c) = \frac{1}{[1 - B_s(c)]} \sum_b y_s(b) \frac{\partial g_s^b}{\partial a_s(c)} \quad (3.16)$$

qui est en fait le coût (ou perte de revenu) dû à l'écoulement d'un appel supplémentaire de classe c sur l'arc s (les résultats numériques que nous avons obtenus semblent indiquer que les $r_s(c)$ sont négatifs), et le revenu net marginal d'un arbre de type k pour la classe c comme

$$R_m^k(c) = w^k(c) + \sum_s r_s(c) \mathcal{I}_{s,(k,c,m)}. \quad (3.17)$$

$R_m^k(c)$ représente l'accroissement de revenu produit si on connecte un appel de classe c de plus sur l'arbre (k, c, m) . On peut voir que ce dernier est le revenu $w^k(c)$ duquel

on soustrait la somme des coûts induits des arcs appartenant à l'arbre en question. Pour des raisons d'ordre pratique, nous introduisons la quantité

$$\bar{R}_m^k(c) = A^k(c)Q_m^k(c)R_m^k(c) \quad (3.18)$$

comme étant le revenu net marginal pondéré de l'arbre (k, c, m) .

Dans la condition d'optimalité (3.15), deux cas peuvent se présenter:

$$\alpha_m^k(c) > 0 \Rightarrow u_m^k(c) = 0 \Leftrightarrow v^k(c) = \bar{R}_m^k(c) \quad (3.19)$$

ou bien

$$u_m^k(c) > 0 \Leftrightarrow v^k(c) > \bar{R}_m^k(c) \Rightarrow \alpha_m^k(c) = 0. \quad (3.20)$$

Ainsi, la condition d'optimalité veut que tous les revenus nets pondérés $\bar{R}_m^k(c)$ soient égaux pour les arbres qui écoulent un flot pour un type donné et que les arbres ayant des revenus nets pondérés plus petits ou égaux à ceux des arbres écoulant du flot n'écoulent pas de flot. On voit ici une généralisation de la condition d'optimalité dans le cas de communications point-à-point [40] où on utilise des chemins à la place des arbres. Notons que nous avons aussi le cas dégénéré $\alpha_m^k(c) = 0$ et $u_m^k(c) = 0$ et alors, on a $v^k(c) = \bar{R}_m^k(c)$ ce qui veut dire qu'un arbre n'écoulant pas de flot peut aussi avoir un revenu égal à ceux des arbres écoulant du flot.

3.2.4 Calcul des multiplicateurs

Puisque les coûts induits r sont des combinaisons linéaires des multiplicateurs y , les valeurs de ces multiplicateurs du point fixe sont données par la résolution d'un système linéaire, dans la mesure où on connaît les valeurs des trafics et des blocages. Ces multiplicateurs donnent la condition d'optimalité par rapports aux probabilités

de blocage **B**. Après un calcul élaboré [40], on obtient

$$y_r(b) = - \sum_{k,m} A^k(b) \alpha_m^k(b) \frac{Q_m^k(b)}{[1 - B_r(b)]} \mathcal{I}_{r,(k,b,m)} \times \left\{ w^k(b) + \sum_{s \neq r} r_s(b) \mathcal{I}_{s,(k,b,m)} \right\}. \quad (3.21)$$

On peut voir dans cette équation les contributions de tous les couples (k, m) au multiplicateur d'un arc donné. Seuls les types d'appel qui ont des arbres utilisant cet arc vont contribuer au multiplicateur. Cette contribution comporte deux parties: la première provient du revenu brut de l'arc alors que la seconde provient des coûts induits sur les autres arcs de l'arbre.

3.3 Synthèse de réseaux

On peut écrire le problème de synthèse comme la maximisation de la valeur nette du réseau, constituée de la différence entre les revenus produits par le réseau et son coût de construction. On obtient alors le problème

$$\min_{\alpha, N} z = \sum_s C_s(N_s) - \sum_{k,c} w^k(c) \bar{A}^k(c) \quad (3.22)$$

$$\sum_m \alpha_m^k(c) = 1 \quad (v^k(c)) \quad (3.23)$$

$$\alpha_m^k(c) \geq 0 \quad (u_m^k(c)) \quad (3.24)$$

$$B_s(c) = g_s^c(\mathbf{a}_s, \mathbf{W}, N_s) \quad (y_s(c)) \quad (3.25)$$

$$L^k(c) \leq \bar{L}^k(c) \quad (x^k(c)) \quad (3.26)$$

auquel nous avons incorporé les équations du point fixe. $C_s(N_s)$ est le coût d'installation d'une capacité N_s sur l'arc s . $\bar{L}^k(c)$ est une borne définissant la probabilité de

perte maximale (QS) pour le couple (k, c) . On obtient pour l'acheminement optimal les conditions

$$v^k(c) = u_m^k(c) + A^k(c) \times \left[Q_m^k(c) \left(w^k(c) + \sum_s r_s(c) \mathcal{I}_{s,(k,c,m)} \right) - L^k(c) \xi^k(c) \right] \quad \forall m \quad (3.27)$$

$$A^k(c) \xi^k(c) = x^k(c). \quad (3.28)$$

Pour des raisons pratiques, nous introduisons le revenu net d'un arbre

$$\overline{\overline{R}}_m^k(c) = \overline{R}_m^k(c) - L^k(c) x^k(c) \quad (3.29)$$

comme étant la différence entre le revenu net marginal pondéré $\overline{R}_m^k(c)$ et le coût dû aux appels perdus $L^k(c) x^k(c)$. On voit que le revenu net d'un arbre est modifié par l'ajout du coût rattaché à la perte d'appels pour le type k . La condition d'équilibre s'exprime toujours par l'égalité des revenus nets pondérés des arbres utilisés, mais cette fois, le revenu net comprend le coût des appels perdus. Nous avons deux cas:

$$\alpha_m^k(c) > 0 \Rightarrow u_m^k(c) = 0 \Leftrightarrow v^k(c) = \overline{\overline{R}}_m^k(c) \quad (3.30)$$

ou

$$u_m^k(c) > 0 \Leftrightarrow v^k(c) > \overline{\overline{R}}_m^k(c) \Rightarrow \alpha_m^k(c) = 0. \quad (3.31)$$

Ainsi, la condition d'optimalité veut que les arbres écoulant du flot positif aient des revenus $\overline{\overline{R}}_m^k(c)$ égaux et que les arbres ayant des revenus $\overline{\overline{R}}_m^k(c)$ inférieurs n'écoulent pas de flot. Ce résultat est similaire à celui trouvé pour l'acheminement excepté que ici, nous avons $\overline{\overline{R}}_m^k(c)$ au lieu de $\overline{R}_m^k(c)$. Notons qu'on a aussi le cas dégénéré où $\alpha_m^k(c) = 0$ et $u_m^k(c) = 0$ et alors, on a $v^k(c) = \overline{\overline{R}}_m^k(c)$.

On peut aussi calculer les équation d'optimalité pour les variables N_s , qui sont les équations de dimensionnement optimal. On obtient

$$\frac{dC_s}{dN_s} = \sum_c y_s(c) \frac{\partial g_s^c}{\partial N_s} \quad (3.32)$$

qui est une équation séparable par faisceau, mais qui comporte un couplage inter-classe. Les multiplicateurs y sont donnés par

$$y_r(b) = -\frac{1}{[1 - B_r(b)]} \sum_{k,m} A^k(b) \alpha_m^k(b) Q_m^k(b) \mathcal{I}_{r,(k,b,m)} \times \quad (3.33)$$

$$\left[w^k(b) + \sum_{s \neq r} r_s(b) \mathcal{I}_{s,(k,b,m)} + \xi^k(b) \right].$$

Une fois qu'on a les multiplicateurs y , on peut déterminer les coûts induits des arcs et les revenus nets des arbres.

3.4 Algorithmes d'optimisation pour un ensemble d'arbres donné

Dans cette section, nous proposons deux algorithmes pour résoudre les problèmes d'acheminement et de dimensionnement en utilisant des méthodes d'optimisation classiques, pour des ensembles d'arbres donnés. Dans la section 3.4.1, nous proposons un algorithme basé sur la méthode du gradient projeté pour la résolution du problème d'acheminement et dans la section 3.4.2, nous proposons un algorithme basé sur la méthode de relaxation Lagrangienne pour la résolution du problème de synthèse. Ces méthodes nous serviront ensuite à résoudre le problème de sélection d'arbres lorsque ceux-ci ne sont pas connus.

3.4.1 Algorithme d'optimisation du partage de charge

Nous voulons résoudre le problème (3.7–3.9) que nous écrivons

$$\min_{\alpha} z(\alpha) = - \sum_{k,c} w^k(c) A^k(c) \quad (3.34)$$

$$\sum_m^M \alpha_m^k(c) = 1 \quad \forall k, c \quad (3.35)$$

$$-\alpha_m^k(c) \leq 0 \quad \forall k, c, m \quad (3.36)$$

où α est le vecteur dont les composantes sont les coefficients de partage de charge. La matrice correspondant aux contraintes (3.35–3.36) qu'on note A est une matrice ayant $(1 + M)KC$ lignes et MKC colonnes, où C, K et M sont respectivement le nombre de classes, le nombre de types d'appel et le nombre d'arbres pour le partage de charge.

Nous supposons que l'ensemble $S^k(c)$ de tous les arbres candidats au partage de charge pour le type k est connu pour chaque type et que le nombre d'arbres dans $S^k(c)$ est $M \forall k$. Le but ici est de trouver un algorithme qui optimise les coefficients de partage de charge α connaissant les $S^k(c)$. Puisqu'on a un problème non linéaire avec contrainte linéaires de la forme

$$\min_x f(x)$$

$$A_1 x = b_1$$

$$A_2 x \leq b_2.$$

où dans notre cas, A_1 correspond aux contraintes (3.35) et A_2 aux contraintes (3.36) et les variables sont les coefficients de partage de charge α , on peut utiliser les méthodes du gradient réduit, du gradient projeté [67] ou de Frank et Wolfe [33]

(détournement de flots) pour le résoudre. Parmi ces méthodes, nous avons retenu la méthode du gradient projeté car elle peut être adéquatement généralisée au cas où les ensembles $\mathcal{S}^k(c)$ ne sont pas connus à priori (voir chapitre 4).

3.4.1.1 Méthode du gradient projeté

L'algorithme de gradient projeté consiste, à une itération donnée, à

1. définir l'ensemble $\Omega(\alpha)$ des contraintes actives et former le sous espace M_W correspondant à ces contraintes. Les contraintes actives sont les contraintes d'égalité (3.35) et les contraintes d'inégalité (3.36) saturées. On forme alors la sous matrice A_q dont les lignes correspondent aux contraintes actives.
2. Calculer la matrice de projection $P = I - A_q^T(A_q A_q^T)^{-1} A_q$ sur le sous espace M_W , et le vecteur déplacement $d = -P \nabla z(\alpha)^T$ qui est la projection de ∇z dans ce sous espace.
3. Si $d \neq 0$, optimiser le pas de déplacement β suivant la direction d de sorte qu'on minimise z dans cette direction, tout en respectant les contraintes. On calcule alors le nouveau vecteur α par $\alpha \leftarrow \alpha + \beta d$.
4. Si $d = 0$, on a trouvé un minimum local dans le sous espace défini par M_W . On calcule alors le vecteur dual

$$\lambda = -(A_q A_q^T)^{-1} A_q \nabla z(\alpha)^T.$$

Deux cas peuvent se présenter:

- (a) Toutes les composantes de λ correspondant aux contraintes d'inégalité actives sont non négatives, les conditions de Kuhn-Tucker sont satisfaites

et on a ainsi un optimum local du problème (3.34–3.36). L'algorithme s'arrête.

- (b) Sinon, on enlève de la matrice A_q la ligne correspondant à l'inégalité qui, à son tour, correspond à la composante du vecteur λ la plus négative puisque c'est cette opération qui donne la plus forte décroissance de z . La contrainte correspondant à cette ligne est relaxée et sort de l'ensemble de contraintes actives. Avec les nouvelles contraintes actives, on forme le nouveau $\Omega(\alpha)$. L'algorithme reprend alors au point 2.

Il faut noter ici que le calcul de λ nécessite la connaissance de toutes les contraintes du problème (tous les arbres). Ceci ne présente pas de difficulté particulière tant que M n'est pas trop grand. Nous verrons dans le chapitre 4 ce qu'il y a lieu de faire si ce n'est pas le cas.

3.4.1.2 Mise en œuvre

En pratique, on n'a pas à faire de produit matriciel pour évaluer le vecteur déplacement d et la structure des contraintes nous permet de calculer d beaucoup plus simplement. En effet, la matrice A peut être vue comme une matrice diagonale de sous-matrices A_j , chacune correspondant à un couple (k, c) :

$$A = \begin{bmatrix} A_1 & 0 & 0 & \dots & 0 \\ 0 & A_2 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & A_{KC} \end{bmatrix}$$

avec

$$A_j = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & -1 \end{bmatrix}.$$

La matrice A_j correspondant au couple (k, c) est une matrice $(M + 1) \times M$. La première ligne correspond à la contrainte d'égalité (3.35) pour le couple (k, c) . Les M lignes restantes correspondent aux contraintes d'inégalité (3.36).

La matrice de projection peut aussi être vue comme une matrice composée de sous-matrices P_j , chacune correspondant à un couple (k, c) . On a

$$P = \begin{bmatrix} P_1 & 0 & 0 & \dots & 0 \\ 0 & P_2 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & P_{Kc} \end{bmatrix}.$$

Supposons qu'à une itération donnée, pour le couple (k, c) , $M - p_{k,c}$ contraintes d'inégalité sont actives. On définit alors l'ensemble $W_c^k(\alpha)$ regroupant ces contraintes. Après réorganisation de la matrice P_j , on a, pour l'indice j

$$P_j = \begin{bmatrix} 0 & . & \dots & . & 0 \\ 0 & . & \dots & . & 0 \\ . & . & . & . & . \\ 0 & 1 - \frac{1}{p_{k,c}} & -\frac{1}{p_{k,c}} & . & -\frac{1}{p_{k,c}} \\ . & . & . & . & . \\ 0 & -\frac{1}{p_{k,c}} & . & -\frac{1}{p_{k,c}} & 1 - \frac{1}{p_{k,c}} \end{bmatrix}.$$

La composante (k, c, m) du gradient projeté s'écrit alors

$$\frac{\partial z}{\partial \alpha_m^k(c)} + \frac{1}{p_{k,c}} \sum_{i \notin \Omega_c^k(\alpha)} \frac{\partial z}{\partial \alpha_i^k(c)}$$

et on peut calculer le nouveau vecteur α par

$$\alpha_m^k(c) \leftarrow \alpha_m^k(c) - \beta \frac{\partial z}{\partial \alpha_m^k(c)} + \frac{\beta}{p_{k,c}} \sum_{i \notin \Omega_c^k(\alpha)} \frac{\partial z}{\partial \alpha_i^k(c)} \quad \forall m \notin \Omega_c^k(\alpha).$$

On peut facilement remarquer que

$$d = 0 \Leftrightarrow \frac{\partial z}{\partial \alpha_i^k(c)} = \frac{\partial z}{\partial \alpha_j^k(c)} \quad \forall k, c; i, j \notin \Omega_c^k(\alpha) \quad (3.37)$$

qui est bien la règle d'équilibre des revenus nets pondérés marginaux des arbres écou-
lant du flot (3.30) pour l'acheminement optimal tel que décrit dans la section 3.2.3.

On peut aussi remarquer que

$$\frac{\partial z}{\partial \alpha_j^k(c)} \geq \frac{1}{p_{k,c}} \sum_{i \notin \Omega_c^k(\alpha)} \frac{\partial z}{\partial \alpha_i^k(c)} \quad \forall j \in \Omega_c^k(\alpha) \quad (3.38)$$

qui est aussi une condition d'arrêt de l'algorithme, est équivalente à la condition d'optimalité (3.31) donnée dans la section 3.2.3 où nous avons dit que les arbres ayant des revenus nets pondérés plus petits que ceux des arbres écoulant du flot n'écoulent pas de flot.

D'autre part, nous avons jugé que pour le problème qui nous intéresse, l'optimisation du pas de déplacement β est très coûteuse en termes de temps de calcul car on doit évaluer la fonction objectif (il faudrait résoudre le point fixe) un nombre très élevé de fois pour trouver ce déplacement optimal. Nous faisons plutôt un déplacement évalué de sorte que le nouveau vecteur α satisfasse toutes les contraintes tout en améliorant la fonction objectif.

Après avoir évalué le nouveau vecteur α , on recalcule une solution du point fixe ce qui correspond à un point fixe résolu à chaque itération de l'algorithme du gradient projeté; puisque nous n'optimisons pas le pas de déplacement β , le point fixe n'est pas résolu lors de la minimisation de la fonction objectif dans la direction

de descente ce qui nous permet de faire beaucoup d'économie de temps de calcul. On évalue aussi la fonction objectif z ainsi que les critères d'arrêt. Les critères d'arrêt sont la convergence de la fonction objectif (ce qui est équivalent à un déplacement nul selon tous les $\alpha_m^k(c)$) et la satisfaction des conditions de Kuhn-Tucker.

À la fin, on peut évaluer les multiplicateurs $y_s(c)$, les revenus nets pondérés $\bar{R}_m^k(c)$ ainsi que les coûts induits $r_s(c)$. La résolution du système linéaire pour trouver les $y_s(c)$ se fait à l'aide de MATLAB. Le reste du programme est écrit en C. L'appel de MATLAB se fait d'une manière automatique à la fin de l'optimisation de l'acheminement.

3.4.2 Algorithme de dimensionnement

Nous supposons que nous connaissons les coûts de dimensionnement $C_s(N_s)$ et nous supposons que ceux-ci sont linéaires c.-à-d. $C_s(N_s) = C_s N_s$. L'algorithme proposé est un algorithme de relaxation lagrangienne. La fonction duale est calculée par relaxation des contraintes de perte (3.26) et le problème dual est résolu par la méthode du sous-gradient. Pour cela, nous définissons la fonction duale $\phi(\mathbf{x})$ du problème (3.22–3.26) par

$$\begin{aligned} \phi(\mathbf{x}) &= - \sum_{k,c} x^k(c) \bar{L}^k(c) + \\ &\quad \min_{\alpha, N} \left[\sum_{k,c} x^k(c) L^k(c) + \sum_s C_s N_s - \sum_{k,c} w^k(c) \bar{A}^k(c) \right] \\ \sum_m \alpha_m^k(c) &= 1 \quad \forall k, c \\ -\alpha_m^k(c) &\leq 0 \quad \forall k, c, m \\ -N_s &\leq 0 \quad \forall s \end{aligned}$$

On résout alors le problème dual en maximisant $\phi(\mathbf{x})$:

$$\begin{aligned} \max_{\mathbf{x}} \phi(\mathbf{x}) \\ -x^k(c) \leq 0 \quad \forall k, c. \end{aligned}$$

Pour s'assurer de l'optimalité des solutions obtenues, nous vérifions la condition d'optimalité donnée par l'équation (3.32). Aussi, dans le cas où on a une seule classe d'appel, nous avons vérifié la relation donnée par l'équation (3.39) où nous avons supprimé l'indice de classe.

$$r_s = -\frac{dC_s}{dN_s} \left(\frac{dN_s}{d\bar{a}_s} \right)_{B=Ct}. \quad (3.39)$$

Cette relation signifie qu'à la solution optimale, le coût induit d'un arc est exactement l'augmentation du coût de dimensionnement d'un arc requise pour écouler un appel supplémentaire à probabilité de blocage constante sur cet arc [38], qui est une conséquence de la définition des coûts induits et des conditions d'optimalité.

3.5 Conclusion

Nous avons présenté une formulation mathématique du problème de synthèse de réseaux notamment, les problèmes d'acheminement et de dimensionnement. La résolution de ces problèmes permet d'évaluer les différentes quantités importantes pour l'analyse du problème de synthèse notamment, les coûts induits des arcs et les coefficients de partage de charge. Étant donné la complexité du modèle, seule une résolution numérique est envisageable.

Nous avons proposé deux algorithmes pour optimiser l'acheminement et le dimensionnement, en utilisant des méthodes classiques d'optimisation. Ces algorithmes

sont efficaces tant du point de vue de la rapidité que de la stabilité de calcul (nous donnons dans le chapitre 4 une évaluation de la qualité de ces algorithmes).

Maintenant qu'on a ces outils, nécessaires pour optimiser l'acheminement et le dimensionnement pour des ensembles d'arbres donnés, nous pouvons aborder le problème d'optimisation de la sélection d'arbres. En effet, les algorithmes proposés dans cette section sont très utiles voire indispensables pour comprendre le problème de sélection d'arbres.

Par exemple, en optimisant l'acheminement on peut savoir si un arbre donné est un bon candidat ou pas en considérant le coefficient de partage de charge qui lui correspond. Si ce coefficient est très faible, ceci voudrait dire qu'on a un mauvais candidat. Ainsi, on peut chercher un critère qui fait en sorte qu'un arbre donné est bon candidat ou pas, comme on va le voir dans les deux chapitres suivants.

CHAPITRE 4

Sélection d'arbres pour l'acheminement

Dans ce chapitre, nous donnons une description du problème de sélection d'arbres dans le contexte de l'acheminement des appels. Dans la section 4.1, nous présentons une formulation mathématique de ce problème. Dans la section 4.2, nous décrivons une étude que nous avons effectuée pour évaluer l'impact de la sélection d'arbres sur les performances du réseau et nous donnons un critère simple pour la sélection d'arbres, dans le contexte d'acheminement d'appels. Dans la section 4.3, nous proposons un algorithme basé sur la méthode du gradient projeté et sur la méthode de génération de colonnes pour résoudre le problème de sélection d'arbres de façon exacte et nous présentons ensuite une méthode heuristique simple pour résoudre ce problème. Ces deux algorithmes ont fait l'objet de l'article [72] et la méthode heuristique a fait l'objet de l'article [74].

4.1 Formulation du problème de sélection d'arbres

Dans [40], on a souligné que pour les techniques d'acheminement d'appels utilisant un ensemble fixe de routes, tel que le partage de charge, il n'y a pas de choix naturel de candidats pour le point-à-multipoint. Ainsi, la question de la sélection d'arbres devient primordiale, et comporte trois aspects:

1. Comment générer les arbres connaissant la source et les destinations. Nous appelons ceci le problème de *génération* d'arbres.
2. Comment construire des ensembles constituant les candidats pour l'acheminement. Ceci est le problème de *sélection* d'arbres.
3. Une fois les candidats connus, comment optimiser le *choix* d'arbres (l'acheminement) afin de maximiser le revenu.

Pour répondre à ces questions, nous utilisons le modèle proposé dans le chapitre 3 et dans [40]. Dans ce modèle, on considère qu'on a un réseau à commutation de circuits virtuels composé exclusivement de commutateurs dupliceurs (*multicast switches*). On suppose aussi que le réseau écoule seulement des connexions point-à-multipoint du type diffusion. On suppose que l'acheminement est un simple partage de charge tel que décrit dans le chapitre 3. Enfin, on suppose qu'on connaît les demandes de trafic qui représentent les taux d'arrivée d'appels d'une source donnée vers un ensemble donné et fixe de destinations.

Le problème de sélection d'arbres dans le contexte d'acheminement peut alors être énoncé comme suit:

Pour une matrice de trafic donnée et pour un réseau donné, avec des liens de capacités connues, trouver la paire (S, α) optimale constituée de

l'ensemble d'arbres \mathcal{S} et de l'acheminement α qui maximise le revenu généré par les connexions.

Dans cet énoncé, l'acheminement et la sélection d'arbres doivent être optimisés simultanément. Malheureusement, cette condition rend toute analyse mathématique du problème très difficile voire, impossible. C'est pourquoi nous allons essayer le plus possible de séparer les deux problèmes au cours de l'analyse théorique. Cependant, lors de l'implantation sous forme d'algorithme d'optimisation de sélection d'arbres, les deux problèmes doivent être traités simultanément. Une analyse mathématique détaillée du problème d'acheminement avec partage de charge est donnée dans le chapitre 3 et dans [40]. La formulation mathématique du problème sélection d'arbres est alors

$$\min_{\alpha, \mathcal{S}} z = - \sum_{k, c, m} w^k(c) A^k(c) \alpha_m^k(c) Q_m^k(c) \quad (4.1)$$

$$\sum_m \alpha_m^k(c) = 1 \quad (4.2)$$

$$\alpha_m^k(c) \geq 0. \quad (4.3)$$

Dans cette formulation, nous n'avons pas limité le nombre d'arbres qui peuvent être utilisés par type d'appel et nous supposons donc que $\mathcal{S}^k(c)$ peut contenir n'importe quel arbre qui permet de connecter les appels de type k . Notons qu'en pratique, le nombre d'arbres pouvant être utilisés peut être très grand et il est impossible d'énumérer tous les arbres pouvant appartenir à $\mathcal{S}^k(c)$.

Avant de présenter la démarche proposée pour résoudre le problème de sélection d'arbres, une question importante est de savoir dans quelle mesure la sélection d'arbres a un impact sur les performances du réseau ou si, peu importe le choix

d'arbres utilisés pour connecter les appels, l'impact sur les performances du réseau est négligeable. Dans le cas où cette sélection d'arbres aurait un impact important, on voudrait alors trouver un critère pour la sélection d'arbres et ensuite modéliser le problème et développer une méthode systématique simple pour le résoudre.

4.2 Impact de la sélection d'arbres sur l'acheminement

Dans cette section, nous voulons illustrer dans quelle mesure la sélection d'arbres est importante. Avant de faire cela, nous rappelons les définitions proposée dans [40] et dans [70] qui sont pertinentes à cette illustration.

4.2.1 Définitions

Dans [40], on définit les notions suivantes:

1. Pour chaque arc s , on définit un *coût induit* $r_s(c)$ qui représente la perte de revenu due à l'augmentation marginale de congestion causée par l'acceptation d'un appel supplémentaire sur l'arc s .
2. Pour chaque arbre, on définit $\bar{R}_m^k(c)$ comme étant le *revenu net marginal pondéré* généré par un appel supplémentaire connecté sur l'arbre. Il s'agit du revenu brut de l'arbre duquel on soustrait la somme des coûts induits des arcs appartenant à l'arbre en question, pondéré par la probabilité de connexion.
3. La règle d'acheminement optimal consiste à choisir les coefficients de partage de charge de telle façon que pour chaque type d'appel, les revenus nets pondérés

des arbres écoulant du trafic sont tous égaux et plus grands que les revenus nets pondérés des arbres n'écoulant pas de trafic.

À cause de la propriété 2, il a été suggéré dans [40] que toutes autres choses étant par ailleurs égales, les arbres de Steiner seraient en désavantage parce qu'ils contiennent plus d'arcs que les arbres simples tels que les arbres de recouvrement. Ainsi, on pourrait s'attendre à ce que les arbres de recouvrement jouent le même rôle dans le contexte du point-à-multipoint que les arcs directs dans le contexte du point-à-point. En d'autres mots, le nombre d'arcs dans l'arbre serait une règle simple pour sélectionner les arbres. Nous voulons étudier cette suggestion et évaluer l'impact de la sélection d'arbres sur les performances du réseau pour éventuellement trouver un critère pour la sélection d'arbres.

Nous avons effectué une étude numérique approfondie de l'impact de la sélection d'arbres sur l'acheminement [70]. Dans cette étude, nous avons supposé que les ensembles d'arbres sélectionnés $\mathcal{S}^k(c)$ sont donnés et fixes, et que la métrique utilisée pour générer les arbres est la distance euclidienne entre les nœuds. En effet, notre objectif étant de savoir si oui ou non la sélection d'arbres est importante, la métrique importe peu dans ce cas. Dans [70], nous définissons les notions importantes qui suivent.

- Un arc *coûteux* est un arc dont le coût induit $r_s(c)$ est élevé (négatif avec une grande valeur absolue) par rapport à ceux des autres arcs. Les arcs coûteux ont une influence considérable sur le choix optimal des arbres. Plus un arbre donné a des arcs coûteux, moins il est susceptible de constituer un bon choix pour écouler du trafic.

- Un arc *commun* est un arc que l'on trouve dans plusieurs arbres qui sont utilisés pour acheminer les appels point-à-multipoint de types différents. Un arc commun a une forte chance d'être coûteux.
- Un arc *critique* est un arc que l'on trouve dans tous les arbres candidats d'un type d'appel donné. Un arc critique est en général très coûteux et affecte considérablement le revenu du type d'appel pour lequel il est critique. Un tel arc affecte aussi considérablement l'acheminement optimal des *autres* types d'appels, pour lesquels cet arc n'est pas critique, lorsque au moins un arbre correspondant à ces autres types d'appel utilise cet arc.

Exemple 4.2.1 Considérons le réseau de $N = 9$ nœuds de la figure 4.1. Nous supposons que nous avons une capacité $N_s = 9$ sur les arcs afin d'avoir une QS acceptable (de l'ordre de 1% de probabilité de perte). Notons que nous avons supposé que tous les arcs ont la même capacité et ce, afin de simplifier l'étude. On peut aussi bien utiliser des arcs de capacités différentes sans toutefois affecter les résultats. Un arc peut être identifié par la paire de nœuds (i, j) qu'il connecte ou par un indice approprié s (ici, on a $i < j$ et $s = iN - (i + 1)(i + 2)/2 + j + 1$). Dans cet exemple, nous considérons une seule classe et un seul type d'appel ($K = 1$) et nous allons supprimer l'indice de classe.

Nous supposons que la source est le nœud 1 et que les destinations sont les nœuds $\{3, 4, 5, 7, 8\}$. Nous supposons que $A^1 = 4$ et que $w^1 = 5$ et ce, en supposant que chaque destination génère une unité (nous supposons que les w^k peuvent facilement être convertis en dollars).

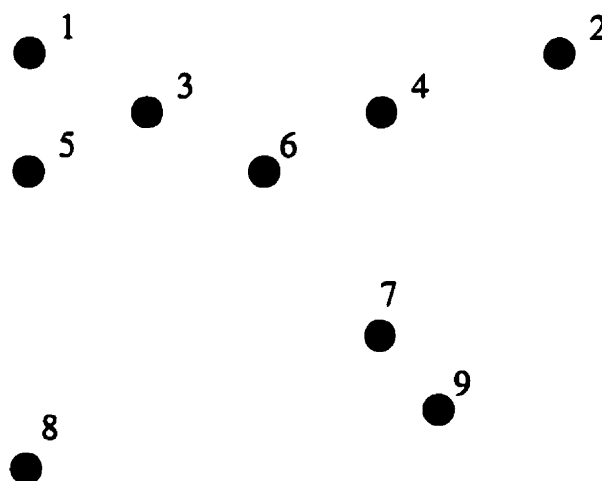


Figure 4.1: *Topologie du réseau.*

Nous considérons trois arbres candidats pour connecter les appels, un arbre des plus courts chemins (étoile) (arbre numéro 1), un arbre de recouvrement de coût minimum (MST) (numéro 2) et un arbre de Steiner (numéro 3) en utilisant la distance euclidienne comme métrique. Ce choix d'arbres est donné dans la figure 4.2. Nous utilisons l'algorithme décrit dans la section 3.4.1.1 pour optimiser l'acheminement et nous obtenons les résultats du tableau 4.1. La probabilité de perte des appels vaut $L^1 = 0.0139$ et le revenu est $|z| = 19.72$.

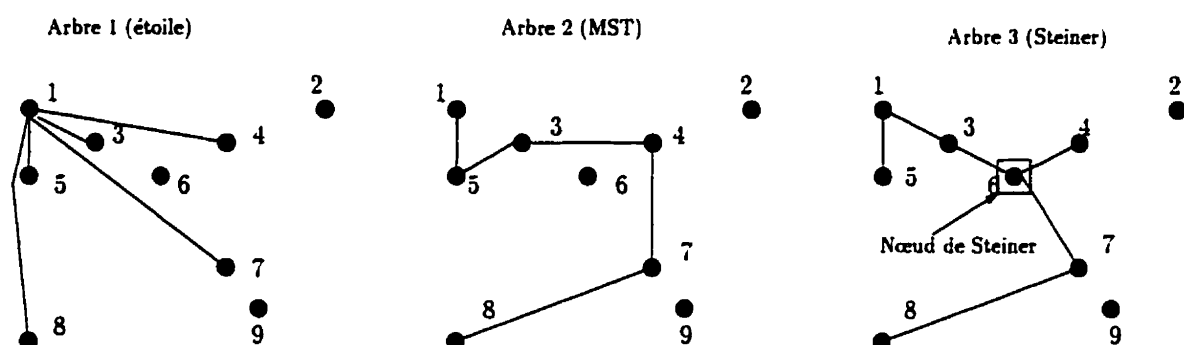


Figure 4.2: *Choix d'arbres pour un seul type d'appel.*

Tableau 4.1: *Résultats du partage de charge pour un seul type d'appel.*

m	α_m^1	Q_m^1	L_m^1	$\partial z / \partial \alpha_m^1$	\bar{R}_m^1
1	0.438	0.986	0.0139	-0.3334	18.46
2	0.438	0.986	0.0139	-0.3334	18.46
3	0.124	0.985	0.0141	-0.3333	18.43

Tableau 4.2: *Variables d'arcs avec un seul type d'appel.*

s	(i, j)	\bar{a}_s	B_s	r_s
1	(1,3)	2.21	0.000387	-0.0130
2	(1,4)	1.72	0.000068	-0.0025
3	(1,5)	3.94	0.013298	-0.2996
5	(1,7)	1.72	0.000068	-0.0025
6	(1,8)	1.72	0.000068	-0.0025
15	(3,4)	1.72	0.000068	-0.0025
16	(3,5)	1.72	0.000068	-0.0025
17	(3,6)	0.48	0.000000	-0.0000
22	(4,6)	0.48	0.000000	-0.0000
23	(4,7)	1.72	0.000068	-0.0025
30	(6,7)	0.48	0.000000	-0.0000
33	(7,8)	2.21	0.000387	-0.0130

D'après les résultats du tableau 4.1, on serait tenté de dire que l'arbre de Steiner ne semble pas être un bon choix puisque le coefficient de partage de charge correspondant est très faible ($\alpha_3^1 = 0.124$). Cette conclusion est hâtive car comme on va le voir plus tard dans cet exemple, c'est un autre facteur qui influence le choix de l'arbre, en l'occurrence, la présence ou non, dans l'arbre, d'arcs communs.

En effet, comme on peut le voir dans le tableau 4.2, les arcs (1, 5), (1, 3) et (7, 8) ont des coûts induits sensiblement élevés par rapport aux autres arcs utilisés. L'arc (1, 5), qui a le coût induit le plus élevé, est présent dans les trois arbres. C'est un arc critique qui ne peut être évité. Les arcs (1, 3) et (7, 8) sont présents dans l'arbre de Steiner. Ces deux arcs ont le même coût induit (et la même probabilité de blocage aussi). L'étoile contient seulement l'arc (1, 3) et l'arbre de recouvrement contient seulement l'arc (7, 8). Ainsi, l'étoile et l'arbre de recouvrement sont parfaitement identiques en terme de revenu net. On peut vérifier dans le tableau 4.1 que ces deux arbres ont les mêmes coefficients de partage de charge. Ces deux arbres sont nettement préférables à l'arbre de Steiner car celui-ci contient deux fois plus d'arcs coûteux que les deux premiers.

Maintenant, nous voulons mettre en évidence le phénomène d'influence des arcs communs sur le choix optimal des arbres. Soit un nouveau type d'appel dont le nœud source est le nœud 6 et les destinations sont $\{1, 2, 4, 5, 7, 9\}$. Nous proposons un choix d'arbres similaire que ci-dessus soit, une étoile, un MST et un arbre de Steiner. Ce choix d'arbres est donné dans la figure 4.3. La demande de trafic est $A^1 = 5$ et le revenu d'un appel est $w^1 = 6$ puisqu'il y a maintenant six nœuds destinataires. Le nombre de circuits requis est $N_s = 11$ sur tous les arcs pour avoir une QS acceptable.

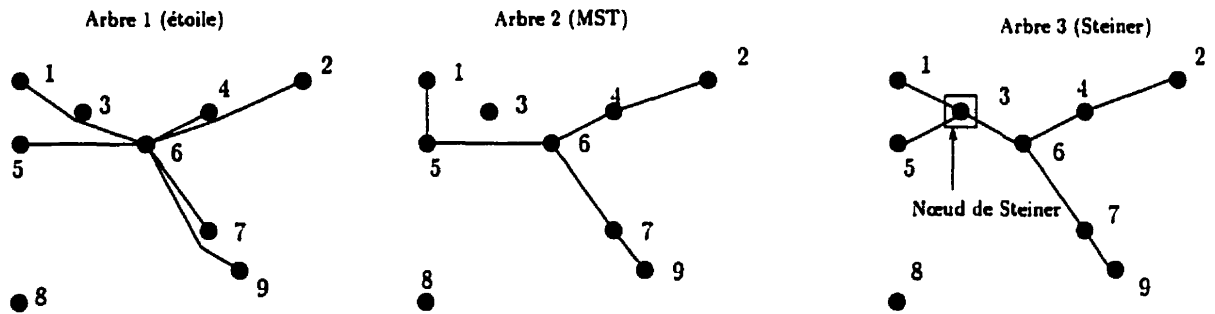


Figure 4.3: *Choix d'arbres pour le nouveau type d'appel.*

Tableau 4.3: *Résultats du partage de charge pour le nouveau type d'appel.*

m	α_m^1	Q_m^1	L_m^1	$\frac{\partial z}{\partial \alpha_m^1}$	\overline{R}_m^1
1	0.453	0.984	0.0159	-0.3333	27.01
2	0.135	0.983	0.0161	-0.3333	26.98
3	0.412	0.984	0.0159	-0.3334	27.01

Les résultats du partage de charge sont donnés dans le tableau 4.3 et ceux donnant les variables d'arcs dans le tableau 4.4. Le revenu est 29.52. La probabilité de perte d'un appel est $L^1 = 0.0159$.

Notons par ailleurs qu'on peut avoir une idée sur la précision des résultats en comparant les \bar{R}_m^1 dans le tableau 4.3 qui doivent être égaux d'après l'équation d'optimalité (3.15). Si on considère seulement trois chiffres significatifs, on remarque que cette condition est satisfaite.

Tableau 4.4: *Résultats pour les arcs avec le nouveau type d'appel.*

s	(i, j)	\bar{a}_s	B_s	r_s
1	(1,3)	2.02	0.000008	-0.0004
3	(1,5)	0.66	0.000000	-0.0000
4	(1,6)	2.22	0.000018	-0.0010
9	(2,4)	2.69	0.000092	-0.0046
11	(2,6)	2.22	0.000018	-0.0010
16	(3,5)	2.02	0.000008	-0.0004
17	(3,6)	2.02	0.000008	-0.0004
22	(4,6)	4.92	0.007888	-0.2494
26	(5,6)	2.89	0.000166	-0.0080
30	(6,7)	4.92	0.007888	-0.2494
32	(6,9)	2.22	0.000018	-0.0010
34	(7,9)	2.69	0.000092	-0.0046

On remarque dans ce cas qu'une faible fraction des appels a été envoyée sur l'arbre de recouvrement. L'arbre de Steiner reçoit près de 41% des appels (voir les α

dans le tableau 4.3). Le choix optimal d'arbres ne semble donc pas être lié au fait que l'arbre soit de Steiner ou pas. C'est plutôt le nombre d'arcs communs qui influence ce choix. En effet, dans ce cas, l'arbre de Steiner utilise moins d'arcs communs que l'arbre de recouvrement. Ce dernier contient les arcs communs $(4, 6)$, $(6, 7)$ et $(5, 6)$ (qui sont coûteux aussi (voir tableau 4.4)) qui sont aussi présents dans l'étoile, tandis que l'arbre de Steiner n'utilise pas l'arc $(5, 6)$ d'où, une préférence pour l'arbre de Steiner plutôt que l'arbre de recouvrement. Notons que les arcs $(2, 4)$ et $(7, 9)$ sont communs au MST et à l'arbre de Steiner et qu'ils ne sont pas utilisés par l'étoile qui reçoit alors légèrement plus de trafic que l'arbre de Steiner (45%).

Exemple 4.2.2 Dans cet exemple, nous considérons trois types d'appel (une seule classe) que nous décrivons dans le tableau 4.5. Les valeurs des revenus w^k sont calculées en supposant que chaque destination génère une unité et que pour le type 3, on a deux unités par destination car on a une longue distance de la source vers les destinations.

Nous utilisons le réseau de $N = 9$ nœuds avec les arcs correspondants donné dans la figure 4.4. Nous supposons que nous avons une capacité $N_s = 10$ sur les 23 arcs du réseau.

Nous utilisons les arbres candidats de la figure 4.5. Nous avons généré pour chaque type d'appel un arbre des plus courts chemins (étoile), un arbre de recouvrement (*MST*) et un arbre de Steiner. Ainsi, nous avons $M = 3$ arbres candidats pour connecter les appels d'un type donné et $K = 3$ types d'appels en tout.

Nous utilisons l'algorithme décrit dans la section 3.4.1.1 pour optimiser l'acheminement. Les résultats pour l'acheminement optimal sont donnés dans le tableau 4.6. Les résultats pour les variables d'arcs optimales sont donnés dans le tableau 4.7.

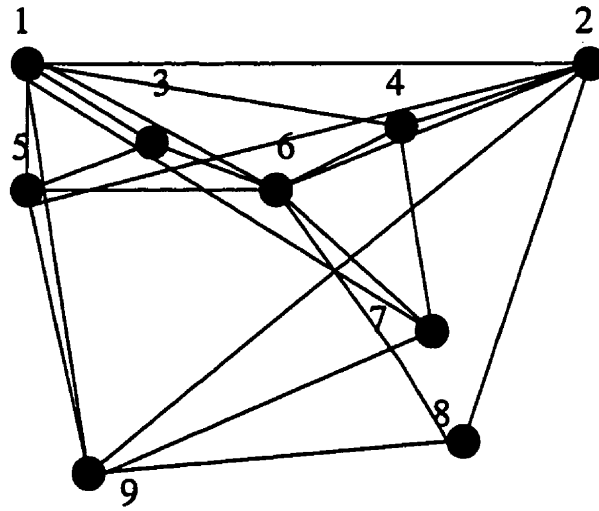


Figure 4.4: *Topologie du réseau.*

Tableau 4.5: *Données.*

k	source	destinations	w^k	A^k
1	1	$\{3,4,5,7,8\}$	5.0	4.0
2	6	$\{1,2,4,5,7,9\}$	6.0	5.0
3	2	$\{1,5,6,8,9\}$	10.0	2.0

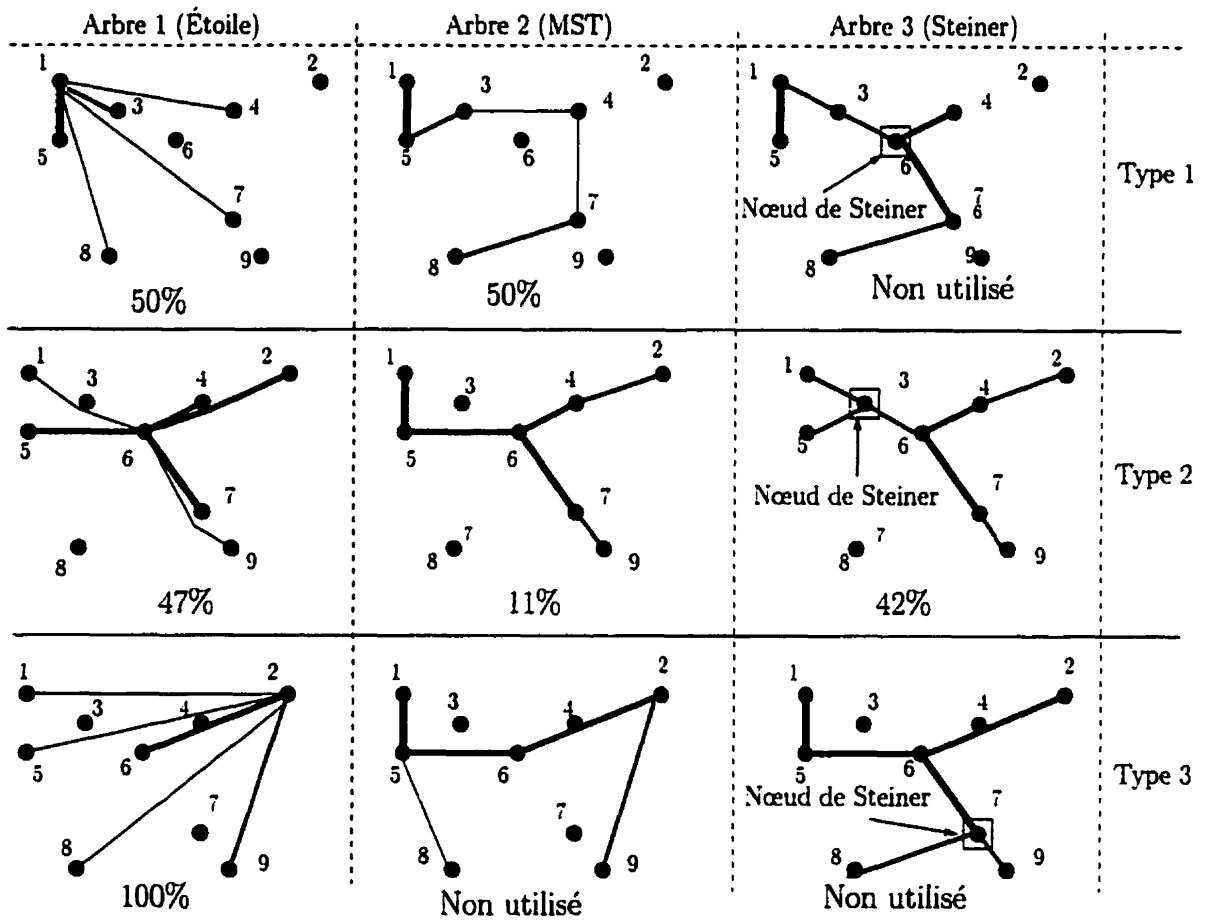


Figure 4.5: Arbres candidats et acheminement optimal (les valeurs en pourcent représentent les coefficients de partage de charge α_m^k). L'épaisseur d'un arc est proportionnelle au nombre d'arbres qui l'utilisent.

Tableau 4.6: *Acheminement optimal générant un revenu total de 68.3.*

k	L^k	Arbre	m	α_m^k	L_m^k	$\partial z / \partial \alpha_m^k$	\overline{R}_m^k
1	0.0160	étoile	1	0.502	0.0161	-0.368	17.90
		MST	2	0.497	0.0159	-0.368	17.92
		Steiner	3	0.000	0.0474	-0.262	13.95
2	0.0414	étoile	1	0.472	0.0403	-0.332	22.77
		MST	2	0.113	0.0433	-0.334	22.87
		Steiner	3	0.414	0.0422	-0.332	22.96
3	0.0081	étoile	1	1.000	0.0081	-0.351	19.14
		MST	2	0.000	0.0193	-0.337	18.32
		Steiner	3	0.000	0.0352	-0.310	17.15

Tableau 4.7: *Variables d'arcs optimales pour les arcs coûteux ($|r_s| \geq 0.14$).*

s	(i, j)	\bar{a}_s	B_s	r_s
1	(1,3)	3.96	0.0051	-0.15
3	(1,5)	4.47	0.0108	-0.29
11	(2,6)	4.25	0.0079	-0.34
16	(3,5)	3.94	0.0050	-0.14
22	(4,6)	4.79	0.0160	-0.44
30	(6,7)	4.79	0.0161	-0.44

On peut voir d'après le tableau 4.7 que nous avons six arcs coûteux. Ces arcs influencent considérablement l'acheminement optimal. En effet, comme on peut le voir dans la figure 4.5, les arbres qui utilisent plusieurs de ces arcs ne sont pas utilisés (n'écoulent pas de trafic). Ces arbres sont l'arbre 3 pour le type 1 et les arbres 2 et 3 pour le type 3. Comme on peut le voir dans le tableau 4.6, les revenus nets pondérés de ces arbres sont inférieurs à ceux des arbres de même type qui écoulent du trafic. On peut ainsi vérifier que la condition d'optimalité (3.15) est satisfaite.

Aussi, on peut remarquer que le choix optimal d'arbres ne dépend pas du fait que l'arbre soit de Steiner ou pas mais plutôt, il dépend du nombre d'arcs coûteux dans les arbres. Par exemple, dans le tableau 4.6, on peut constater que l'arbre de Steiner du type 2 reçoit plus de trafic que l'arbre de recouvrement (MST) du même type ($\alpha_3^2 > \alpha_2^2$). Ainsi, même si l'arbre de Steiner a plus d'arcs que l'arbre de recouvrement, il reçoit plus de trafic ce qui veut dire que ce n'est pas le nombre d'arcs dans l'arbre qui importe dans le choix optimal (acheminement) mais plutôt, le nombre d'arcs communs dans l'arbre. En effet, comme on peut le voir dans la figure 4.5, l'arbre de recouvrement pour le type 2 utilise l'arc (1, 5) qui est utilisé par six arbres en tout et l'arc (5, 6) qui est utilisé par quatre arbres en tout. L'arbre de Steiner quant à lui utilise les arcs (1, 3) (utilisé par trois arbres), l'arc (3, 5) (utilisé par deux arbres) et l'arc (3, 6) (utilisé par deux arbres) (les arcs (2, 4), (4, 6), (6, 7) et (7, 9) sont communs à l'arbre de Steiner et à l'arbre de recouvrement de type 2 et n'influencent pas le choix). C'est le fait que les arcs de l'arbre de Steiner soient utilisés par un petit nombre d'arbres (ils sont peu communs) qui a fait en sorte qu'il reçoit plus de trafic.

D'autre part, la QS du type 2 est très mauvaise (plus de 4% de perte d'appels) comme on peut le voir dans le tableau 4.6 (colonne L^k). En effet, comme on peut le voir dans la figure 4.5, pour ce type d'appel, on a deux arcs critiques, les arcs (4,6) et (6,7). Comme ces arcs sont très coûteux (voir tableau 4.6), ils ne peuvent être évités et cela résulte en une performance relativement mauvaise pour ce type d'appel.

On peut améliorer la performance du réseau en évitant 1) l'utilisation d'arcs critiques et 2) les arcs communs. Nous avons alors sélectionné les arbres en faisant en sorte que 1) les arbres du même type soient disjoints (aucun arc en commun) et 2) un arc donné ne peut être utilisé qu'au plus par deux arbres en considérant tous les types d'appel. Nous avons alors dû limiter le nombre d'arbres à deux par type d'appel car il est impossible d'avoir plusieurs arbres disjoints. Cependant, nous nous sommes assurés d'utiliser exactement le même nombre d'arcs en tout (23) que dans le cas où nous avons la sélection d'arbre de la figure 4.5 afin que la comparaison soit équitable. Nous obtenons la sélection d'arbres de la figure 4.6.

Avec cette nouvelle sélection d'arbres, nous obtenons un revenu de 69.02 et $L^k \leq 0.029 \forall k$ (à comparer à un revenu de 68.3 et $L^k \leq 0.041 \forall k$ avec la sélection d'arbres de la figure 4.5). Ainsi, nous avons amélioré la performance du réseau en sélectionnant adéquatement les arbres et ce, même si nous avons moins d'arbres candidats par type d'appel (on est passé de $M = 3$ à $M = 2$) ce qui illustre l'importance de bien choisir les arbres. De plus, comme on peut le constater dans la figure 4.6, le choix optimal d'arbre (valeurs de α_m^k en pourcent) ne donne aucune préférence particulière pour un arbre ou pour un autre, dans la mesure où chaque arbre d'un type d'appel donné reçoit pratiquement 50% du trafic.

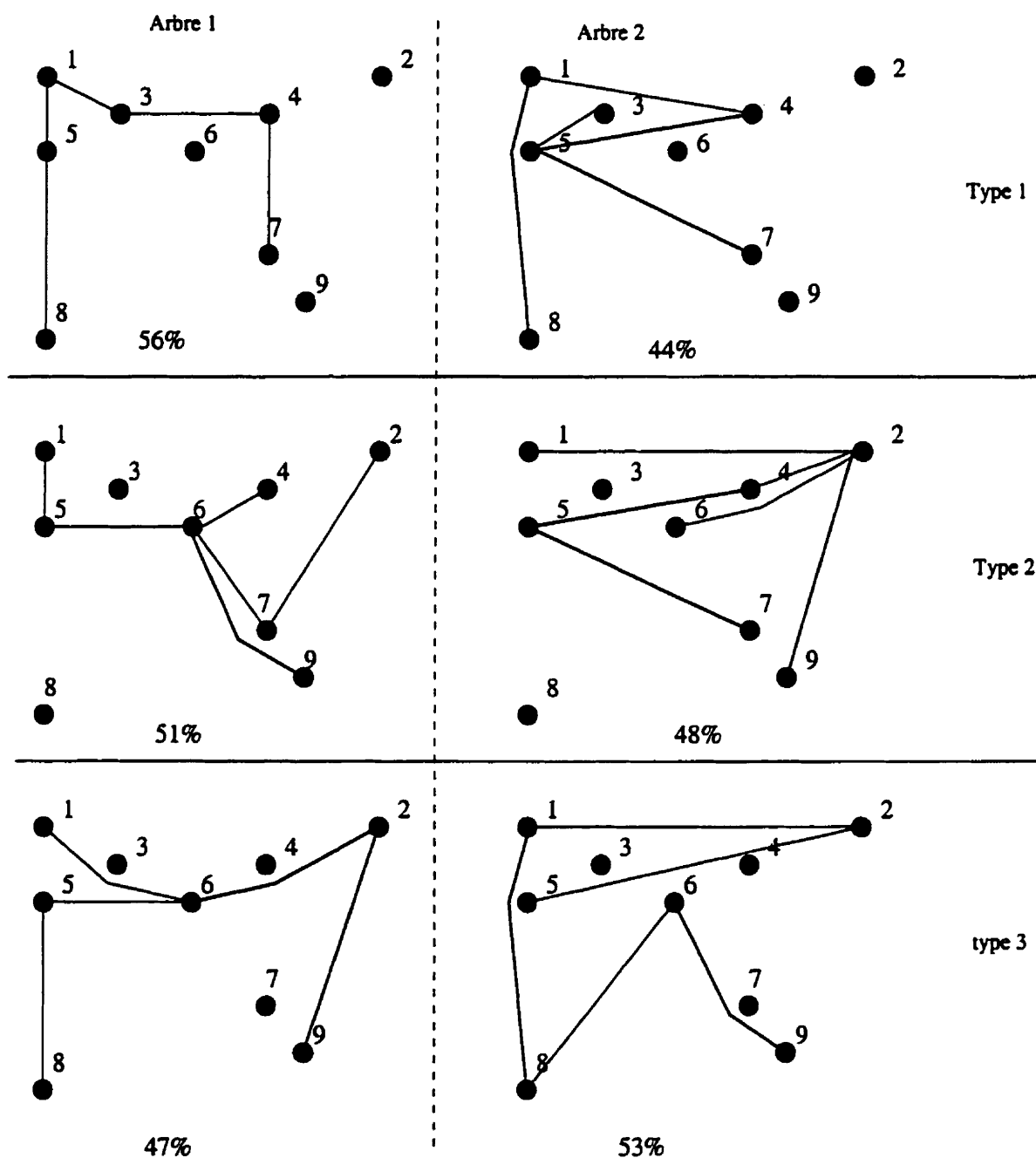


Figure 4.6: Sélection d'arbres et acheminement optimal avec un petit nombre d'arcs en commun (les valeurs en pourcent donnent le partage de charge optimal).

Exemple 4.2.3 Nous considérons maintenant un exemple où nous avons deux classes d'appels. Aux trois types d'appel décrits dans le tableau 4.5, nous ajoutons deux types d'appels de débit deux fois supérieur à celui de ces trois types d'appels. Nous avons donc $W(1) = 1$ et $W(2) = 2$ (pour les autres paramètres, voir tableau 4.8). Nous voulons vérifier que les résultats obtenus dans l'exemple 4.2.2 sont aussi valables dans le contexte du multi-débit.

Pour chacun des deux types d'appel de classe 2 décrits dans le tableau 4.8, nous générons trois arbres, une étoile, un arbre de recouvrement et un arbre de Steiner de coûts minimum. Dans cet exemple, nous avons supposé que les revenus $w^k(c)$ sont proportionnels au nombre de destinations par type et au débit des connexions. Afin de tenir compte du trafic additionnel ajouté dans le réseau, nous avons augmenté les capacités des arcs à $N_s = 17 \forall s$.

Les arbres candidats et l'acheminement optimal sont donnés dans la figure 4.7. Comme nous pouvons le constater dans cette figure, les arbres ayant beaucoup d'arcs communs (ceux représentés avec des lignes épaisses) n'écoulent pas de trafic ou écoulent peu de trafic ce qui généralise le résultat sur l'influence des arcs coûteux et critiques sur le choix optimal d'arbres. En effet, pour le type 1 de classe 2, nous avons deux arcs critiques (arcs (3, 5) et (3, 6)) ce qui dégrade considérablement la QS de ce type d'appel.

Une remarque intéressante que l'on peut aussi faire dans le contexte du multi-débit est que le coût induit semble être proportionnel au débit de la classe, comme on peut le voir dans le tableau 4.9. Ceci laisse suggérer que *le prix à payer pour écouler un appel supplémentaire sur un arc donné est proportionnel au débit de l'appel en question*. Ceci suggère aussi que les revenus $w^k(c)$ devraient être proportionnels au

débits $W(c)$. Notons que cette conclusion serait due à l'approximation de Labourdette [64] que nous utilisons pour trouver les probabilités de blocage sur les arcs selon laquelle, asymptotiquement, les appels multi-débit se comportent comme si les circuits étaient saisis et libérés indépendamment [64].

Il nous est difficile de trouver une bonne sélection d'arbres par essais et erreur avec plusieurs types d'appel. Cela justifie la nécessité de développer des procédures systématiques pour sélectionner les arbres dans le cas général. Comme on va le voir dans l'exemple 4.3.2, nous pouvons considérablement améliorer la performance du réseau pour les données de cet exemple en sélectionnant de bons arbres.

Tableau 4.8: *Données pour le multi-débit.*

c	k	source	destinations	$w^k(c)$	$A^k(c)$
1	1	1	{3,4,5,7,8}	5.0	4.0
	2	6	{1,2,4,5,7,9}	6.0	5.0
	3	2	{1,5,6,8,9}	10.0	2.0
2	1	3	{2,5,6,8,9}	10.0	4.0
	2	7	{2,3,4,5}	8.0	3.0

4.2.2 Résumé

Le résultat le plus important que l'on peut dégager de cette étude numérique peut être résumé comme suit:

Pour le problème d'acheminement, le critère le plus important sur lequel la sélection d'arbres doit se baser semble être la présence ou non d'arcs

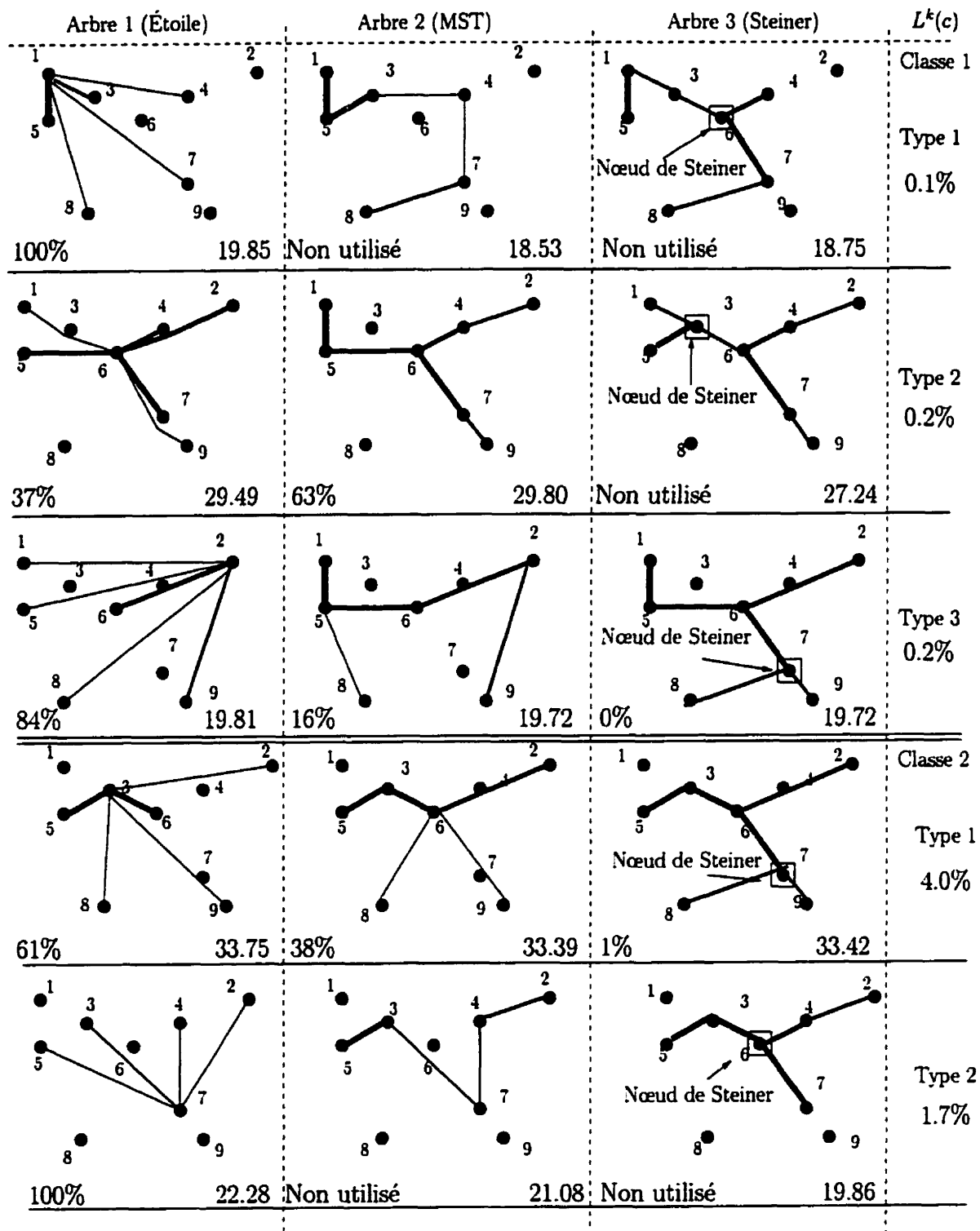


Figure 4.7: Arbres candidats pour le multi-débit. Pour chaque arbre, nous donnons le coefficient de partage de charge en pourcent (coin gauche en bas) et le revenu net pondéré (coin droit en bas). Le revenu total est 131.9.

Tableau 4.9: Variables d'arcs optimales pour les arcs coûteux pour le multi-débit ($|r_s(c)| \geq 0.15$).

c	s	(i, j)	$a_s(c)$	$B_s(c)$	$r_s(c)$
1	12	(2,6)	3.87	0.0019	-0.078
	16	(3,5)	0.	0.	-0.275
	17	(3,6)	0.	0.	-0.275
2	12	(2,6)	1.50	0.0054	-0.156
	16	(3,5)	3.91	0.0178	-0.560
	17	(3,6)	3.91	0.0178	-0.560

communs dans les arbres utilisés pour acheminer les appels des différents types d'appel. L'utilisation d'arbres avec peu d'arcs en commun réduit la chance d'avoir des arcs coûteux et on peut augmenter ainsi le revenu des appels. D'autre part, si nous utilisons des arbres avec des arcs en commun, à l'acheminement optimal, le trafic est partagé plus ou moins uniformément entre les arbres. De plus, il faut éviter l'utilisation d'arcs critiques.

Nous avons ainsi testé la suggestion à l'effet de sélectionner des arbres avec un petit nombre d'arcs et nous avons constaté que cette règle n'est pas valide en général. Au contraire, un point clé dans la sélection de bons candidats est le fait que les arbres aient ou pas des arcs communs, un résultat directement lié aux coûts induits des arcs. Nous avons montré que pour avoir un bon revenu, il faut avoir des arbres disjoints et ce, indépendamment du fait que ces arbres soient des arbres de recouvrement ou des arbres de Steiner. Ceci va faire en sorte que le trafic soit *dispersé* sur un maximum

possible d'arcs et on évite ainsi d'avoir des arcs trop chargés comme c'est le cas des arcs critiques.

Maintenant qu'on sait que la sélection d'arbres a un impact important sur la performance du réseau, on a besoin d'outils pour générer et sélectionner les arbres de façon systématique. Ces outils doivent nous permettre d'optimiser et l'acheminement et les ensembles d'arbres candidats $S^k(c)$.

4.3 Algorithmes de sélection d'arbres

On veut savoir comment résoudre le problème de sélection d'arbres décrit par les équations (4.1– 4.3) qui sont très complexes et qui impliquent plusieurs problèmes à la fois. Nous décrivons dans cette section deux méthodes pour résoudre ce problème, une méthode exacte basée sur le gradient projeté et la génération de colonnes, qui illustre la complexité du problème, et une méthode heuristique.

4.3.1 Méthode exacte

Pour résoudre le problème (4.1–4.3) de façon exacte, il nous faut une méthode qui n'exige pas la connaissance des arbres à priori. C'est pour cette raison que nous avons fait appel à la méthode du gradient projeté. En fait, pour évaluer le gradient projeté, on a besoin de connaître seulement les arbres dont les contraintes d'inégalité données par l'équation (4.3) ne sont pas saturées. Nous utilisons alors une forme légèrement modifiée de l'algorithme du gradient projeté décrit dans la section 3.4.1.1.

Cette modification concerne le point 4 de l'algorithme où on a trouvé un minimum local dans le sous espace défini par M_W . Dans ce cas, il faut vérifier si on a

intérêt à quitter la surface courante. Il faut donc s'assurer que toutes les composantes de λ correspondant aux contraintes saturées sont non négatives.

Malheureusement, à cause du très grand nombre d'arbres que l'on peut avoir, on ne peut pas énumérer tous les éléments de l'ensemble de contraintes actives (ou saturées) $W_c^k(\alpha)$ afin d'évaluer le vecteur dual λ comme on l'a fait dans la section 3.4.1.1. La difficulté vient donc de la vérification de la condition d'optimalité donnée par l'équation (3.38) pour toutes les contraintes actives. En pratique, le nombre de contraintes actives peut être très grand.

Toutefois, on sait que la contrainte donnée par l'équation (3.38) est équivalente à la condition d'optimalité (3.15) selon laquelle, les revenus nets pondérés $\bar{R}_m^k(c)$ des arbres n'écoulant pas de trafic doivent être inférieurs ou égaux aux revenus nets pondérés des arbres écoulant du trafic. Cette condition peut s'exprimer par

$$u_j^k(c) = \bar{R}_i^k(c) - \bar{R}_j^k(c) \geq 0 \quad \forall j \in \Omega_c^k(\alpha), \quad \forall i \notin \Omega_c^k(\alpha). \quad (4.4)$$

et on sait calculer $\bar{R}_m^k(c)$ connaissant les coûts induits $r_s(c)$, les probabilités de blocage $B_s(c)$ et l'arbre (k, c, m) . On peut ainsi vérifier cette condition dans la mesure où

- si on trouve au moins une paire (k, c) dans $\Omega_c^k(\alpha)$ pour laquelle il existe m' tel que $u_{m'}^k(c) < 0$ (ce qui est équivalent à dire qu'il y a une composante négative dans λ) alors, la contrainte correspondant à l'arbre ainsi généré est relaxée puisque cela revient à dire qu'il existe un arbre pour lequel le revenu net pondéré $\bar{R}_{m'}^k(c)$ est supérieur à celui des arbres qui écoulent actuellement du trafic.

- Sinon, on n'a pas intérêt à quitter la surface courante, les conditions de Kuhn-Tucker sont satisfaites on a donc atteint un minimum local et l'algorithme s'arrête.

L'algorithme consiste alors à générer l'arbre (k, c, m) qui minimise $u_m^k(c)$ ou encore, qui maximise le revenu net pondéré $\bar{R}_m^k(c)$ en utilisant $r_s(c)$ comme longueurs d'arcs ce qui revient à générer un arbre de Steiner de coût pondéré maximum où les coûts des arcs sont donnés par les coûts induits. Cela revient à résoudre le problème suivant pour chaque couple (k, c)

$$\max_{m \in \Omega_c^k(\alpha)} \bar{R}_m^k(c). \quad (4.5)$$

Ainsi, pour chaque paire (k, c) , la contrainte susceptible de quitter $\Omega_c^k(\alpha)$ est obtenue en trouvant la solution du problème (4.5).

Ceci est en fait une méthode de génération de colonnes [40] communément utilisée en programmation linéaire quand on a un très grand nombre de variables [75]. Il s'agit, à chaque itération, d'introduire une colonne qui minimise les coûts réduits. Dans notre cas, une colonne correspond à un arbre par couple (k, c) . Notons qu'en pratique, tous les arbres pour lesquels $u_m^k(c) < 0$ peuvent quitter $\Omega_c^k(\alpha)$ en même temps mais en pratique, on se contente de relaxer une seule contrainte à la fois.

Théoriquement, l'algorithme décrit ci-dessus converge vers un minimum local. Les difficultés qui restent à surmonter sont alors liées à la complexité du calcul. Ces difficultés peuvent être résumées en trois points:

1. Complexité du problème de génération d'arbres de Steiner de coût pondéré maximum. Il faut noter que dans la mesure où les coûts induits sont négatifs (ce qui peut être vérifié numériquement), le problème de génération d'arbres

de Steiner de coût maximum est équivalent au problème de génération d'arbre de Steiner de coût minimum en inversant le signe des coûts induits. La difficulté vient essentiellement de la pondération qui est un problème tout à fait nouveau qui, à notre connaissance, n'a pas été étudié dans la littérature. En effet, d'après l'équation (3.18), nous avons besoin de connaître $Q_m^k(c)$ pour générer les arbres qui maximisent $\bar{R}_m^k(c)$. Cette quantité comprend des produits des probabilités de connexion sur les arcs alors que tous les algorithmes de génération d'arbres de Steiner que nous connaissons utilisent des coûts additifs. Nous avons donc à la fois des coûts additifs et multiplicatifs et nous ne connaissons pas d'heuristiques de génération d'arbres de Steiner applicable à notre problème. En pratique, nous générons d'abord un arbre de Steiner de coût minimum en utilisant les coûts additifs (les $r_s(c)$) seulement (en considérant toutes les combinaisons possibles de nœuds de Steiner) puis, évaluer $Q_m^k(c)$ et $\bar{R}_m^k(c)$ et ensuite, sélectionner l'arbre qui maximise $\bar{R}_m^k(c)$. Néanmoins, nous pouvons supposer que $Q_m^k(c)$ peut être négligé lors de la génération d'arbres dans la mesure où $Q_m^k(c) \cong 1$. En effet, en pratique, les bons arbres doivent avoir une forte probabilité de connexion faute de quoi, ils seraient peu susceptibles de générer plus de revenu que les arbres qui écoulent du trafic. Notons par ailleurs que le problème de génération d'arbres de Steiner de coût minimum dans les graphes est un problème NP-difficile [75, 23, 60] mais il y a des heuristiques pour trouver les arbres de Steiner [84, 36, 98] de coût minimum. Cependant, comme nous voulons une résolution exacte, nous n'avons pas utilisé ces heuristiques.

2. Complexité du calcul du gradient ∇z qui affecte considérablement la convergence de l'algorithme d'optimisation de l'acheminement. En effet, en se basant sur plusieurs exemples numériques, nous avons constaté que la convergence de l'algorithme d'optimisation de l'acheminement peut être très lente si on ne limite pas le choix d'arbres et le nombre d'arbres pour chaque type d'appel. Dans ce cas, le nombre d'itérations requis pour la convergence de l'algorithme d'optimisation de l'acheminement croît considérablement avec le nombre de types d'appel.
3. Complexité du calcul des coût induits à chaque itération de l'algorithme. Ces coûts induits sont calculés en trouvant la solution d'un système linéaire pour les multiplicateurs y . On peut alors s'attendre à ce que le temps de calcul soit très grand si le réseau est dense et contient plusieurs arcs et plusieurs classes d'appels. Par exemple, nous savons que dans le contexte du point-à-point, le calcul des multiplicateurs y est une composante dominante dans le temps de calcul de l'acheminement optimal.

D'autre part, en se basant sur plusieurs exemples numériques, nous avons constaté que la solution finale de cet algorithme dépendait fortement de la solution initiale (une solution désigne une sélection d'arbres avec son acheminement optimal). Les arbres d'une itération donnée sont générés à partir des coût induits calculés pour les arbres de l'itération précédente et cela affecte beaucoup la qualité de la solution optimale obtenue par cet algorithme si on ne choisit pas attentivement la solution initiale. De plus, nous avons constaté que le problème n'est pas convexe dans la mesure où, en partant de solutions initiales différentes, nous obtenons des solutions optimales différentes aussi. Cette non convexité est très forte puisque pour

pratiquement chaque solution initiale que nous avons essayée, nous avons obtenu systématiquement une nouvelle solution optimale.

Ainsi, on a besoin d'une approche plus simple pour trouver de bonnes sélections d'arbres. Dans la section 4.3.2, nous proposons une méthode heuristique où on réduit considérablement la complexité du problème de sélection d'arbres. Cette méthode pourrait servir entre autre à trouver une bonne solution initiale pour la méthode exacte si on voulait en calculer une.

4.3.2 Méthode heuristique

La résolution du problème de sélection d'arbres par la méthode exacte peut prendre beaucoup de temps pour converger surtout à cause du problème de génération d'arbres de Steiner de coût pondéré maximum. On a aussi constaté que cette méthode pouvait donner des optimums locaux de mauvaise qualité à cause notamment de sa forte dépendance de la solution initiale. Nous proposons dans cette section une méthode heuristique qui permet de trouver rapidement une sélection quasi optimale d'arbres. Nous expliquerons plus loin dans cette section pourquoi les solutions obtenues par la méthode heuristique sont quasi optimales. Nous proposons les deux éléments simplificateurs suivants:

- Nous introduisons $\overline{M}^k(c)$ comme étant le nombre maximal d'arbres qu'on peut envisager par type d'appels ($\overline{M}^k(c)$ est une donnée du problème). Nous introduisons également $M^k(c)$, le nombre d'arbres effectivement utilisés pour connecter les appels de type k . Cette quantité doit elle aussi être optimisée. En effet, il n'y aucune raison que ce nombre soit le même pour tous les types d'appel. On a $M^k(c) = |S^k(c)| \leq \overline{M}^k(c)$.

- Les arbres générés pour un type d'appel donné devront faire partie d'un ensemble $\mathcal{T}^k(c)$ fixé à l'avance. Nous supposons que nous pouvons vérifier facilement si un arbre donné appartient à $\mathcal{T}^k(c)$ même si on ne peut pas énumérer tous les éléments de $\mathcal{T}^k(c)$. Ceci pourrait être une contrainte sur le nombre maximal de nœuds de Steiner dans les arbres par exemple. En pratique, il faudra que $\mathcal{T}^k(c)$ soit assez grand pour couvrir un nombre suffisant de solutions admissibles, mais pas trop grand pour éviter que l'algorithme ne devienne trop complexe.

Nous nous proposons alors de résoudre le problème suivant:

$$\min_{\alpha, S} z = - \sum_{k, c, m} w^k(c) A^k(c) \alpha_m^k(c) Q_m^k(c) \quad (4.6)$$

$$\sum_{m=0}^{M^k(c)} \alpha_m^k(c) = 1 \quad (4.7)$$

$$\alpha^k(c)_m \geq 0 \quad (4.8)$$

$$M^k(c) \leq \overline{M}^k(c) \quad (4.9)$$

$$S^k(c) \subset \mathcal{T}^k(c) \quad (4.10)$$

où nous avons introduit la contrainte (4.9) qui impose une limite $\overline{M}^k(c)$ au nombre d'arbres pouvant appartenir à l'ensemble $S^k(c)$ et la contrainte (4.10) afin de limiter le nombre de choix possibles d'arbres.

4.3.2.1 Approximation du revenu

Dans cette section, nous donnons une approximation du revenu généré par les connexions afin de simplifier le problème de sélection d'arbre. Nous notons \hat{X} une approximation d'une variable X si elle correspond à une borne supérieure et \check{X} si elle correspond à une borne inférieure. Nous avons fait en sorte que les approximations

satisfassent $\hat{z} \geq z$ et $\hat{L}^k(c) \geq L^k(c)$ sans pour autant que les bornes ne soient trop larges. Ceci garantit que si $\hat{L}^k(c)$ satisfait les contraintes de QS, alors $L^k(c)$ les satisfait aussi ($L^k(c) \leq \hat{L}^k(c) \leq \bar{L}^k(c)$).

Dans la mesure où nous utilisons un partage de charge statique et que nous considérons une seule tentative de connexion, nous devons avoir de très faibles probabilités de blocage sur les arcs ($B_s(c) \ll 1$) pour avoir une bonne QS [72]. En pratique, pour $L^k(c)$ donné, $B_s(c)$ est de l'ordre $L^k(c)/D$ où D est le nombre moyen de destinations par type d'appel. Par exemple, si $L^k(c)$ est de l'ordre de 0.01 et qu'en moyenne, nous avons $D = 10$, $B_s(c)$ est de l'ordre de 0.001. On peut alors approximer le trafic offert à un arc s de classe c au trafic total qu'on aurait s'il n'y avait pas de blocage sur l'arc. Ainsi, nous approximations $a_s(c)$ par

$$\hat{a}_s(c) = \sum_{k,m} \alpha_m^k(c) A^k(c) \mathcal{I}_s(k, c, m) \quad (4.11)$$

et nous pouvons alors écrire

$$\hat{B}_s(c) = g_c(\hat{a}_s(c), \mathbf{W}, N_s) \quad (4.12)$$

d'où nous pouvons approximer la probabilité de connexion sur un arbre donné par

$$\tilde{Q}_m^k(c) = 1 - \sum_s \hat{B}_s(c) \mathcal{I}_{s,(k,c,m)}. \quad (4.13)$$

La fonction objectif z peut alors être approximée par

$$\hat{z} = - \sum_k A^k(c) w^k(c) + \sum_{k,c,m} A^k(c) w^k(c) \alpha_m^k(c) \sum_s \hat{B}_s(c) \mathcal{I}_{s,(k,c,m)}. \quad (4.14)$$

Expérimentalement, nous avons vérifié que \hat{z} est une bonne approximation de z (de l'ordre de 0.3% plus élevée que z) pour des probabilités de perte d'appels $L^k(c)$ de moins de 1%. De même, nous avons pu vérifier, sur plusieurs exemples où $B_s(c) \ll$

1 $\forall s$, que l'approximation sur les $L^k(c)$ est acceptable (voir exemples numériques ci-dessous et [72]). Minimiser \hat{z} est alors équivalent à minimiser

$$\sum_{k,c,m} A^k(c) w^k(c) \alpha_m^k(c) \sum_s \hat{B}_s(c) \mathcal{I}_{s,(k,c,m)}$$

qui représente une approximation de la perte de revenu due aux blocages sur les arcs. Ainsi, pour des coefficients de partage de charge donnés et pour un type k de classe c donné, un bon candidat serait l'arbre qui minimise la somme des probabilités de blocage de l'arbre

$$\hat{L}_m^k(c) = \sum_s \hat{B}_s(c) \mathcal{I}_{s,(k,c,m)} \quad (4.15)$$

qui est en fait une approximation de la probabilité de perte d'un appel sur l'arbre en question. L'idée est alors basée sur la génération d'arbres qui minimisent $\sum_s \hat{B}_s(c)$ ce qui donne un algorithme générateur de colonnes (une colonne désigne un arbre) qui est nettement plus simple que l'algorithme décrit par le problème (4.5).

4.3.2.2 Heuristique de sélection d'arbres

Nous voulons générer des arbres de recouvrement de coût minimum et/ou des arbres de Steiner de coût minimum avec les $\hat{B}_s(c)$ comme longueurs d'arcs, malgré que l'on sait que la génération d'arbres de Steiner de coût minimum est complexe. En fait, dans l'algorithme que nous allons proposer, nous allons limiter l'ensemble d'arbres possibles $\mathcal{T}^k(c)$ aux arbres de Steiner d'au plus n nœuds de Steiner avec n petit (de l'ordre de 2). Les arbres de Steiner sont générés en faisant une énumération complète de tous les arbres possibles de $\mathcal{T}^k(c)$, en considérant toutes les combinaisons possibles de nœuds de Steiner, et en choisissant celui qui a le coût minimum.

Par ailleurs, les résultats numériques ont montré que la solution initiale affecte considérablement la solution finale. Ainsi, nous voulons être en mesure de générer de

bons arbres dès les premières itérations de l'algorithme de sélection d'arbres. Malheureusement, cela nécessite la connaissance des $\hat{B}_s(c)$ qui ne peuvent être évaluées que si nous avons les arbres alors que ces derniers nécessitent la connaissance des $\hat{B}_s(c)$ afin d'être générés. En pratique, nous calculons les $\hat{B}_s(c)$ au fur et à mesure que nous générons les arbres comme nous allons le décrire dans la procédure donnée ci-dessous.

Nous avons aussi constaté qu'en commençant par générer des arbres pour les types d'appel ayant une forte intensité de trafic (fortes valeurs de $W(c)$ et/ou $A^k(c)$), nous obtenons de meilleurs résultats que si l'ordre dans lequel nous générons les arbres est arbitraire [72]. En effet, comme la solution initiale a un impact important sur la solution finale, il vaut mieux commencer par générer des arbres pour les types d'appel ayant une forte intensité de trafic car au fur et à mesure que nous générons des arbres, le nombre d'arcs non utilisés diminue et on a alors de moins en moins de choix pour générer des arbres disjoints. Alors, si on laisse les types d'appel avec une forte intensité de trafic aux dernières itérations, le fait qu'on ait un choix limité pour générer de bons arbres pour ces types d'appel a de fortes chances de donner de mauvaises solutions.

La procédure de sélection d'arbres est alors la suivante:

1. Arranger les types d'appels dans l'ordre décroissant de $W(c)A^k(c)$.

2. $\hat{a}_s(c) = 0 \forall s, c$, $M^k(c) = 0 \forall k, c$, $M_g = 0$, $k = 1$.

$$(a) \quad M_t = \sum_{k,c} \overline{M}^k(c) - M^k(c).$$

(b) Si $M^k(c) < \overline{M}^k(c)$, $m = M^k(c) \leftarrow M^k(c) + 1$, $M_g \leftarrow M_g + 1$. Sinon, aller à 2f.

- (c) Évaluer $\hat{B}_s(c) = g_c(\hat{a}_s(c) + A^k(c), \mathbf{W}, N_s)$.
 - (d) Générer l'arbre qui minimise $\hat{L}_m^k(c)$ (les arbres doivent appartenir à $\mathcal{T}^k(c)$) (voir procédure ci-dessous).
 - (e) Pour chaque arc s utilisé par le type k de classe c , $\hat{a}_s(c) \leftarrow \hat{a}_s(c) + A^k(c)$.
 - (f) Si $M_g = M_t$, aller à (3). Sinon, $k = k + 1 \pmod{K + 1}$, aller à (2b).
3. Optimiser l'acheminement à l'aide de l'algorithme décrit dans la section 3.4.1.
 4. Pour chaque arbre de \mathcal{S} , si $\alpha_m^k(c) = 0$, enlever l'arbre (k, c, m) de $\mathcal{S}^k(c)$, $M^k(c) \leftarrow M^k(c) - 1$.
 5. Si le revenu augmente (z diminue), $\hat{a}_s(c) = a_s(c) \forall s$, aller à (2a). Sinon, fin.

où M_g et M_t sont des variables entières qui donnent le nombre d'arbres générés à l'itération courante et le nombre d'arbres total qui reste à générer à l'itération courante, respectivement, en considérant tous les types d'appel.

Cet algorithme est en fait une heuristique gloutonne qui fonctionne sur un ensemble donné d'arbres. À chaque itération, on génère un nouvel arbre pour chaque type d'appel. Cet arbre doit appartenir à un ensemble d'arbres bien défini (ici les arbres d'au plus n nœuds de Steiner). Notons que cet ensemble d'arbres peut être ajusté selon la connectivité du réseau (on mesure la connectivité en termes de nombre d'arcs dans le réseau). Plus le réseau est connecté, plus on devrait limiter le nombre maximum de nœuds de Steiner dans les arbres et vice-versa et ce, dans le but de réduire la complexité de l'algorithme d'une part, et d'avoir un nombre suffisamment grand de solutions admissibles, d'autre part. En pratique, nous avons constaté que pour des réseaux moyennement connectés, $n = 2$ était acceptable [72].

Supposons que $\mathcal{T}^k(c)$ contient les arbres d'au plus n nœuds de Steiner et que N est le nombre de nœuds dans le réseau. Un arbre avec exactement i nœuds de Steiner est généré en considérant $\binom{N}{i}$ arbres de recouvrement (ces arbres recouvrent les destinations du type d'appel k et les i nœuds de Steiner) et en sélectionnant l'arbre qui minimise $L_m^k(c)$. Après avoir généré tous les arbres pour $0 \leq i \leq n$, nous sélectionnons l'arbre qui minimise $L_m^k(c)$ qui est celui que nous cherchons dans l'étape 2d de l'algorithme.

Dans la mesure où la complexité de l'algorithme *MST* est $O(N^2)$ où N est le nombre de nœuds du réseau, générer un arbre avec un nœud de Steiner présente une complexité $O(N^{2+1})$ puisqu'on fait appel N fois à l'algorithme *MST* et en général, générer un arbre avec n nœuds de Steiner présente une complexité $O(N^{2+n})$ puisqu'on fait appel $\binom{N}{n}$ fois à l'algorithme *MST* et dans la mesure où $\binom{N}{n}$ peut être réduit en un polynôme en N de degré n . Ainsi, la complexité de l'algorithme de génération d'arbres est $O(N^{n+2})$.

En supposant que $\overline{M} = \max_{k,c} \overline{M}^k(c)$, dans la mesure où on génère un arbre par type d'appel dans chaque itération et où on doit générer au plus \overline{M} arbres par type d'appel et en supposant qu'on a K types d'appel par classe, et C classes d'appel, la complexité de l'algorithme de sélection d'arbres est $O(KC\overline{M}N^{2+n})$. Il est alors évident que n doit être petit et que cette méthode ne s'appliquera à des grands réseaux ($N > 100$) que pour n très petit ($n = 1$).

Les trois avantages principaux de l'algorithme de sélection d'arbres que nous venons de présenter sont 1) il ne nécessite pas une solution initiale (celle-ci est générée automatiquement), 2) il ne nécessite pas la connaissance des coûts induits ce qui simplifie la génération d'arbres et 3) il réduit considérablement la complexité

du problème de sélection d'arbres en limitant le choix d'arbres à un ensemble prédéfini.

4.3.2.3 Qualité de l'heuristique

Une façon de vérifier la qualité de l'heuristique est de la tester avec des données pour lesquelles on connaît la solution optimale. Pour ce faire, on dimensionne un réseau pour une sélection d'arbres générés arbitrairement, afin d'obtenir un réseau avec des capacités optimales pour cette sélection d'arbres. Maintenant, si on cherche les arbres optimaux pour le réseau ainsi obtenu, on doit retrouver ceux qu'on a générés arbitrairement. Cela a été vérifié pour quelques exemples et nous obtenons bien les arbres optimaux désirés sauf pour certains cas particuliers où il a été possible de trouver une autre sélection d'arbres (à cause de la non convexité du problème), mais qui donne sensiblement les mêmes performances obtenues par dimensionnement. D'autre part, nous avons pu vérifier que l'algorithme converge au bout de quelques itérations (en moyenne 6 itérations) peu importe la taille du réseau alors que le temps de calcul dans chaque itération augmente avec la taille du réseau et du nombre de types d'appel [72].

Une autre façon de vérifier la qualité des solutions obtenues par la méthode heuristique est de comparer le revenu obtenu à celui qu'on aurait si les contraintes de QS étaient juste satisfaites. Par exemple, si on admet des probabilités de perte d'appels d'au plus 1% alors, le revenu obtenu par la sélection d'arbres devrait être supérieur ou égal à 99% du revenu total (brut) qu'on obtiendrait si on écoulait tous les appels (sans perte). Cela a été vérifié sur une dizaine d'exemples différents et on a obtenu même mieux que 99% du revenu brut dans la plupart des cas. (On

qualifiera d'*acceptable* une sélection d'arbres pour laquelle le revenu est supérieur ou égal à 99% du revenu total).

Une autre possibilité consiste à établir des bornes inférieures sur le revenu généré par les appels et la QS selon une analyse théorique rigoureuse. Malheureusement, cela constitue une tâche très difficile. La seule borne que nous avons (le revenu brut des appels) est relativement large et ne peut donner qu'une idée sur la qualité des solutions obtenues.

Enfin, une autre façon de vérifier la qualité des solutions obtenues à l'aide de ces procédures de sélection d'arbres consiste à les comparer à des solutions exactes obtenues à l'aide de l'algorithme décrit dans la section 4.3.1 comme nous allons le décrire dans l'exemple 4.3.1. On pourrait aussi comparer nos méthodes avec des méthodes génétiques ou de type tabou tel que décrit dans la section 4.5.

Exemple 4.3.1 Nous utilisons les mêmes données de l'exemple 4.2.2. Nous utilisons l'algorithme de la section 4.3.1 pour résoudre le problème de sélection d'arbres de façon exacte et la méthode heuristique décrite dans la section 4.3.2 pour le résoudre de façon approximative. Nous voulons comparer la performance du réseau pour les sélections d'arbres obtenues par les deux méthodes d'une part et comparer le temps de calcul requis par les deux méthodes, d'autre part.

Notons par ailleurs que nous avons choisi des réseaux ni sous-dimensionnés, ni sur-dimensionnés où il y avait place pour une amélioration de la QS et du revenu par un choix non trivial de l'acheminement. Un acheminement trivial serait obtenu si peu importe l'acheminement, la performance du réseau est acceptable (moins de 1% de perte d'appels).

Pour la méthode heuristique, nous voulons évaluer la qualité des solutions en

fonction du nombre maximum de nœuds de Steiner n et le nombre maximum d'arbres \overline{M} où nous avons supposé que $\overline{M}^k = \overline{M} \forall k$. Les solutions ainsi obtenues sont utilisées comme solution initiales pour la méthode exacte. Notons que dans la mesure où le problème de sélection d'arbres n'est pas convexe, la méthode exacte donne des solutions différentes pour des valeurs initiales différentes.

Tableau 4.10: *Résultats pour différentes valeurs de n et \overline{M} .*

(\overline{M}, n)	Revenu		$\max_k L^k$		temps (min)	
	heuristique	exacte	heuristique	exacte	heuristique	exacte
(1,0)	64.7	69.4 (7% de plus)	.115	.010	0	3
(1,1)	65.8	69.2 (5% de plus)	.076	.013	0	2
(2,0)	69.0	69.8 (1% de plus)	.028	.004	0	7
(2,1)	69.1	69.8 (1% de plus)	.025	.004	1	6
(3,0)	69.5	69.8 (0% de plus)	.018	.003	2	8
(3,1)	69.7	69.8 (0% de plus)	.008	.003	2	5

Les résultats sont donnés dans le tableau 4.10. D'après ces résultats, on constate que plus on augmente n (de 0 à 1) et/ou \overline{M} , plus on améliore la qualité de la sélection d'arbres obtenue par la méthode heuristique. Cependant, nous avons constaté que pour $n \geq 2$ et pour $\overline{M} \geq 4$, nous n'avons plus une amélioration importante de la solution. En fait, pour les données de cet exemple, les solution pour $n = 2$ et pour $\overline{M} = 4$ sont identiques à celles obtenues pour $n = 1$ et $\overline{M} = 3$ (seul le temps de calcul augmente pour $n = 2$ et $\overline{M} = 4$). Ceci veut dire qu'il n'est pas intéressant d'utiliser des arbres de Steiner très grands d'une part et trop d'arbres par type

d'appel d'autre part car dans le premier cas, nous augmentons inutilement le temps de calcul et dans le deuxième cas, il n'est plus possible de générer autant d'arbres disjoints [72].

Nous constatons aussi d'après les résultats du tableau 4.10 que peu importe la solution initiale (obtenue par la méthode heuristique), nous obtenons une sélection d'arbres acceptable avec la méthode exacte. D'autre part, nous constatons que le temps de convergence de la méthode exacte peut être très long. En fait, avec la méthode exacte on a soit un temps très long pour trouver une bonne sélection d'arbres comme c'est le cas pour $(\overline{M}, n) = (2, 0)$ ou un temps relativement court auquel cas, on a un optimum local qui n'est pas nécessairement très bon comme c'est le cas pour $(\overline{M}, n) = (1, 1)$.

La meilleure sélection d'arbres (méthode heuristique) est obtenue pour $(\overline{M}, n) = (3, 1)$. Cette solution est légèrement améliorée par la méthode exacte (surtout pour la QS). En effet, avec la méthode exacte, on peut utiliser autant d'arbres et de nœuds de Steiner qu'on veut (ici, à la solution optimale nous avons jusqu'à six arbres par type d'appel dans la solution exacte) mais puisque la méthode heuristique donne une solution où moins de 0.8% des appels sont perdus, on peut très bien se contenter de cette solution.

Le revenu total (brut) qu'on aurait si tout le trafic était écoulé est de 70.0 ce qui veut dire que si on admet 1% de perte, on devrait avoir 69.3 comme revenu. La solution obtenue avec la méthode heuristique pour $(\overline{M}, n) = (3, 1)$ donne un revenu de 69.7 ce qui est mieux que le minimum exigé. Avec les approximations, nous avons $|\hat{z}| = 69.7$. Remarquons que dans l'exemple 4.2.2 le revenu était de 68.3 ce qui n'est pas acceptable. Les résultats importants sont résumés dans le tableau 4.11.

Tableau 4.11: *Acheminement des appels point-à-multipoint.*

Sélection d'arbres	Revenu	$\max_k L^k$
Réf. (étoile,MST,Steiner)	68.3	.041
Heuristique	69.7	.008
Exacte	69.8	.003

La meilleure sélection d'arbres (méthode heuristique) et l'acheminement optimal correspondant sont donnés dans la figure 4.8. On voit qu'il n'y a pas de préférence particulière dans l'acheminement optimal dans la mesure où tous les arbres sélectionnés ont un petit nombre d'arcs coûteux. Si on compare la performance du réseau avec celle de l'exemple 4.2.2, d'après le tableau 4.11, on voit qu'on a une amélioration du revenu et surtout de la QS. En effet celle-ci passe de $\max_k L^k = 0.041$ dans le cas où la sélection d'arbres est constituée d'une étoile, d'un MST et d'un arbre de Steiner pour chaque type d'appel, à $\max_k L^k = 0.008$ dans le cas où on utilise la méthode heuristique pour sélectionner les arbres et ce, en utilisant le même nombre d'arbres par type d'appel, soit trois arbres. Avec les approximations, nous avons constaté que nous avons pratiquement les même valeurs pour \hat{L}^k et L^k .

Exemple 4.3.2 De même que dans l'exemple 4.3.1, nous allons maintenant sélectionner des arbres à l'aide de la méthode heuristique puis par la méthode exacte mais cette fois, dans le contexte du multi-débit, pour les données de l'exemple 4.2.3. Nous avons fixé le nombre maximum d'arbres à $\overline{M} = 3$ et le nombre maximal de nœud de Steiner à $n = 1$.

Nous obtenons des arbres disjoints dont le choix optimal ne favorise aucun arbre

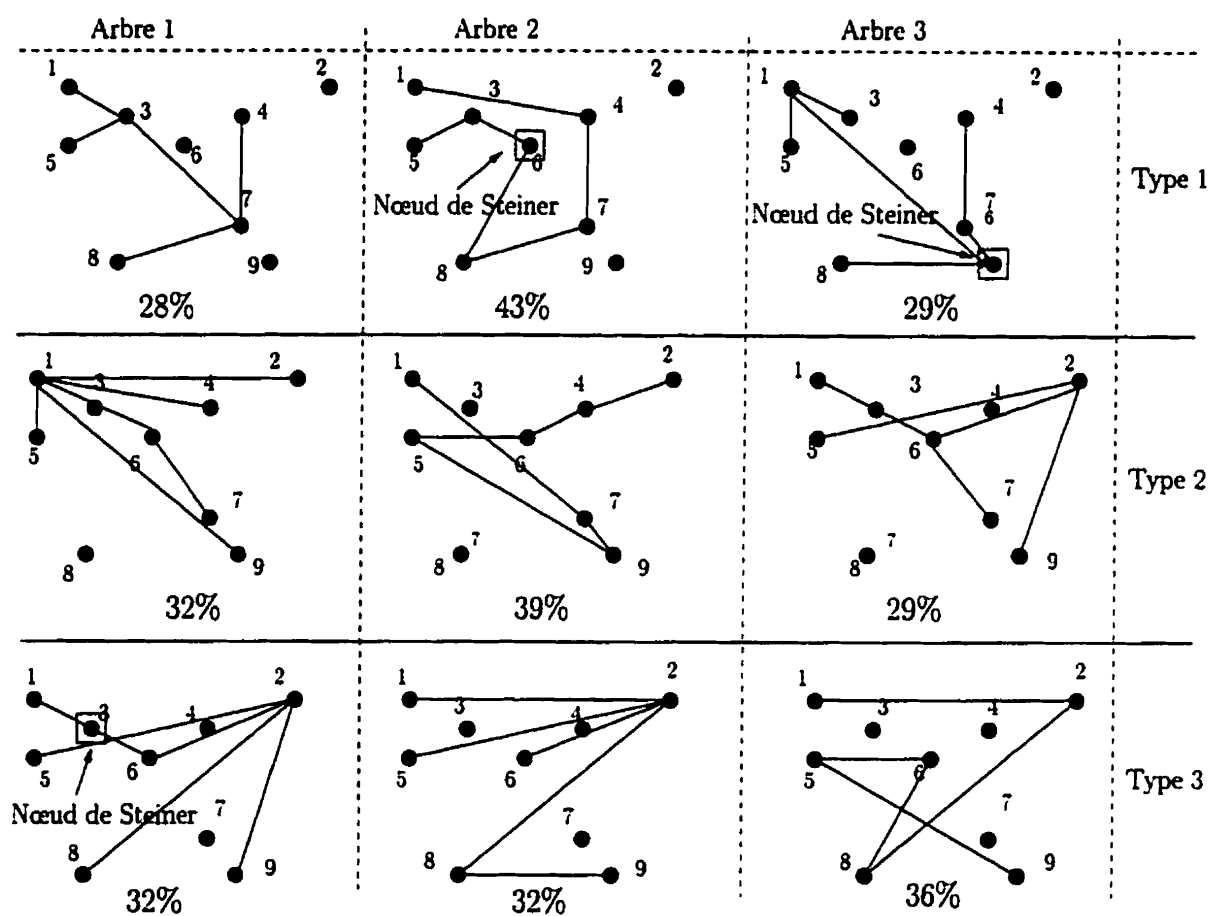


Figure 4.8: Sélection d'arbres avec la méthode heuristique.

Tableau 4.12: *QS pour le multi-débit.*

(k, c)	(1, 1)	(2, 1)	(3, 1)	(1, 2)	(2, 2)
$L^k(c)$	0.000	0.000	0.001	0.005	0.004
$\hat{L}^k(c)$	0.000	0.000	0.001	0.005	0.004

en particulier. Le revenu est de 133.7 ($|\hat{z}| = 133.7$) et la qualité de service est donnée dans le tableau 4.12. Comme nous pouvons le constater, nous avons légèrement amélioré le revenu et nettement amélioré la QS par rapport aux résultats de l'exemple 4.2.3. En effet, dans cet exemple là, le revenu était de 131.9 et on avait $\max_k L^k = 0.04$. Ces résultats sont résumés dans le tableau 4.13.

Tableau 4.13: *Acheminement des appels point-à-multipoint avec du multi-débit.*

Sélection d'arbres	Revenu	$\max_k L^k$
Réf. (étoile, MST, Steiner)	131.9	.040
Heuristique	133.7	.005
Exacte	133.8	.003

4.3.3 Résultats pour d'autres réseaux

Dans cette section, nous voulons évaluer la qualité de l'heuristique en fonction de la taille du réseau (en termes de nombre de nœuds), du nombre de types d'appel et du nombre de classes.

Exemple 4.3.3 Dans cet exemple, nous considérons une seule classe d'appels et nous faisons varier le nombre de nœuds dans le réseau en fixant le nombre de types d'appel K à 10. Pour chaque réseau, nous avons effectué exactement la même étude que dans l'exemple 4.3.1 et nous donnons la meilleure sélection d'arbres obtenues par la méthode heuristique et les valeurs de (\bar{M}, n) qui permettent d'obtenir cette sélection d'arbres. La solution obtenue par la méthode exacte est obtenue en partant d'une solution arbitraire car nous voulons illustrer la différence entre les temps de calcul des deux méthodes. Cependant, nous nous sommes assurés d'obtenir une solution acceptable par la méthode exacte et ce, en commençant par différentes solutions initiales. Les résultats sont donnés dans le tableau 4.14.

D'après ces résultats, nous pouvons constater que le temps de calcul pour la méthode exacte croît très rapidement avec la taille du réseau alors que la croissance est quasi linéaire avec la méthode heuristique. En effet, lorsque N augmente, la complexité du problème de recherche de l'arbre de Steiner croît exponentiellement et si on ne limite pas le nombre maximal de nœud de Steiner, le temps de calcul peut être trop long comme c'est le cas pour $N = 25$ dans le tableau 4.14.

Tableau 4.14: *Résultats pour différents nombre de nœuds pour $K = 10$.*

N	(\bar{M}, n)	$\sum_k w^k A^k$	Revenu		$\max_k L^k$		temps (min)	
			heur.	exacte	heur.	exacte	heur.	exacte
9	(3, 1)	22.0	21.8	21.8	.010	.010	0	2
16	(3, 2)	31.6	31.4	31.4	.009	.008	9	47
25	(3, 2)	46.3	45.9	45.8	.010	.012	27	5816

D'autre part, pour $N = 16$, nous avons constaté que la solution obtenue par la méthode heuristique est un optimum local pour la méthode exacte [72]. En effet, si nous commençons par la solution obtenue par la méthode heuristique, la méthode exacte ne donne aucune amélioration, un résultat qui a aussi été observé sur d'autres réseaux.

Exemple 4.3.4 Dans cet exemple, nous considérons aussi une seule classe mais nous faisons varier le nombre de types d'appel K et nous fixons N à 16. De même, pour chaque réseau, nous avons effectué exactement la même étude que dans l'exemple 4.3.1 et nous donnons la meilleure sélection d'arbres. Les résultats sont donnés dans le tableau 4.15.

Tableau 4.15: *Résultats pour différents nombre de types d'appels pour $N = 16$.*

K	(\overline{M}, n)	$\sum_k w^k A^k$	Revenu		$\max_k L^k$		temps (min)	
			heur.	exacte	heur.	exacte	heur.	exacte
10	(3, 2)	31.6	31.4	31.4	.009	.008	9	47
15	(4, 1)	43.3	42.9	42.9	.010	.010	37	62
20	(3, 2)	51.9	51.5	51.5	.011	.010	67	120

Comme nous pouvons le constater, le temps de calcul de la méthode heuristique et celui de la méthode exacte augmentent quasi linéairement avec le nombre de types d'appel. Ceci est dû au fait que nous nous sommes concentrés sur la simplification des problèmes de génération et sélection d'arbres et non pas sur la simplification du problème d'acheminement. En effet, lorsque K augmente, le nombres de composantes

du vecteur α croît rapidement ce qui complique le calcul de ∇z . Ainsi, il faudrait trouver un moyen pour approximer le calcul de ∇z , ce qui constitue une tâche non triviale.

Exemple 4.3.5 Dans cet exemple, nous fixons N et K et nous faisons varier le nombre de classes. Nous considérons trois cas, le cas où nous avons une classe avec $W(1) = 1$, deux classes avec en plus $W(2) = 2$ et trois classe avec en plus $W(3) = 4$. Nous supposons que nous avons dix types d'appel en tout.

Pour chaque cas, nous sélectionnons les arbres à l'aide de la méthode heuristique puis par la méthode exacte en partant d'une solution initiale arbitraire. Les résultats sont donné dans le tableau 4.16 pour le revenu et dans le tableau 4.17 pour la QS.

Tableau 4.16: *Résultats pour différentes classes d'appel pour $N = 9$ et 10 types d'appel.*

C	(\overline{M}, n)	$\sum_{k,c} w^k(c)A^k(c)$	Revenu		temps (min)	
			heur.	exacte	heur.	exacte
1	(3, 1)	22.0	21.8	21.8	0	2
2	(3, 2)	24.2	24.1	24.1	12	16
3	(3, 2)	48.0	47.8	47.8	18	24

Comme on peut le constater d'après les résultats du tableau 4.16, la méthode heuristique donne des solutions comparables à celles obtenues par la méthode exacte en un temps plus court.

D'autre part, comme on peut le constater dans le tableau 4.17, les probabilités de perte des appels à faible débit sont beaucoup plus petites que celles des appels

Tableau 4.17: $\max_k L^k(c)$ pour des réseau multi-débit.

N	c	Une classe		Deux classes		Trois classes	
		heuristique	exacte	heuristique	exacte	heuristique	exacte
9	1	.010	.010	.003	.003	.001	.002
	2	—	—	.009	.009	.003	.003
	3	—	—	—	—	.007	.007

à haut débit. Ceci est dû au fait que nous n'utilisons pas de réservation de circuits dans notre modèle. Dans ce cas, les grosses capacités des arcs font en sorte que pour les appels de faible débit, peu importe la route utilisée pour connecter ces appels, leur QS est acceptable. Ainsi, nous pourrions simplifier la sélection d'arbre en la considérant seulement pour les appels de haut débit ce qui se ramènerait à considérer une seule classe d'appel.

En effet, dans le cas du multi-débit, le nombre de variables d'arcs ($a_s(c)$ et $B_s(c)$) augmente avec le nombre de classes ce qui complique la résolution du point fixe. En effet, comme on peut le voir dans le tableau 4.16, même pour un petit réseau de 9 nœuds, pour les deux méthodes de sélection d'arbres, on a un temps de calcul très long dans le cas de trois classes d'appel par rapport au cas où on a une classe.

4.3.3.1 Sélection de $\mathcal{T}^k(c)$ et $\overline{M}^k(c)$

Nous avons montré que la qualité de la sélection d'arbres obtenue par la méthode heuristique dépendait considérablement de $\mathcal{T}^k(c)$ et $\overline{M}^k(c)$. La question qui se pose est alors comment trouver la meilleure sélection d'arbres dans un minimum de temps sans avoir à effectuer une étude exhaustive tel que décrit ci-dessus.

Intuitivement, nous savons que nous devons avoir un domaine suffisamment grand afin d'avoir suffisamment de solutions admissibles sans pour autant avoir un domaine trop grand qui augmenterait la chance de trouver de mauvais candidats. En pratique, nous avons constaté que dans un réseau suffisamment connecté, $n = 1$ ou 2 était suffisant. De plus, nous avons constaté qu'avoir $n \geq 3$ n'était pas nécessairement avantageux dans la mesure où dans ce cas, on perd beaucoup de temps à tester des mauvais candidats (de trop gros arbres) et la chance de trouver de mauvais optimums locaux augmente. Cependant, il n'est pas toujours possible d'avoir des arbres d'au plus deux nœuds de Steiner pour connecter tous les types d'appel car cela dépend de la connectivité du réseau. Alors, pour les types d'appel concernés, on n'a pas le choix que de générer de gros arbres de Steiner (avec plus de deux nœuds de Steiner).

D'autre part, pour trouver $\overline{M}^k(c)$, il est important de garder présent à l'esprit que nous devons avoir des arbres avec un petit nombre d'arcs en commun. $\overline{M}^k(c)$ n'a donc pas à prendre des valeurs trop grandes dans la mesure où il est naturellement impossible d'avoir un grand nombre d'arbres disjoints. En pratique, nous avons constaté qu'indépendamment de la taille du réseau, $\overline{M}^k(c) = 3$ ou 4 donnait des résultats acceptables.

4.3.4 Résumé

Nous avons testé l'algorithme basé sur la méthode heuristique sur plusieurs réseaux. Dans tous les cas étudiés, l'algorithme nous a permis d'obtenir rapidement des solutions très satisfaisantes en terme de revenu et qualité de service.

D'autre part, les solutions obtenues par cette méthode heuristique sont comparables en termes de performances à celles obtenues par la méthode exacte. En fait, nous avons constaté que la méthode exacte dépendait beaucoup de la solution initiale et qu'en partant d'une solution arbitrairement choisie, on pouvait obtenir des solutions moins bonnes que celles obtenues par la méthode heuristique [72]. Cela est dû à la non convexité du problème et ainsi, au fait que les solutions obtenues sont des optimums locaux. De plus, pour n et \overline{M} assez grands, nous avons souvent obtenu des solutions exactes identiques à celles obtenues par la méthode heuristique lorsque la solution obtenue par cette méthode heuristique est utilisée comme solution initiale pour la méthode exacte; d'où l'expression *quasi optimales* pour décrire les solutions obtenues par la méthode heuristique [72].

Enfin, nous avons montré qu'il n'est pas nécessaire de considérer de très grands arbres de Steiner dans la mesure où ceux-ci ont de fortes chances de contenir des arcs coûteux et ainsi, de générer de faibles revenus.

4.4 Acheminement en temps réel

L'heuristique que nous avons proposée pour la sélection d'arbre pour le problème d'acheminement dépend de la connaissance à priori des matrices de trafic. En pratique, il peut être difficile de caractériser ces matrices de trafic et nous pouvons avoir de grosses différences entre les valeurs prévues et les valeurs réelles de ces matrices de trafic. De plus, le nombre de types d'appel peut être immense ce qui rend très difficile la connaissance de ces types d'appel. Alors, on se demande que faire si en pratique, nous ne connaissons pas ces matrices de trafic. Par exemple,

que se passe-t-il si un appel arrive alors qu'il ne correspond à aucun type d'appel pré-défini?

Une fois que nous avons sélectionné les arbres, nous avons les valeurs des probabilités de blocage sur les arcs. Comme nous l'avons expliqué dans la section 4.3.2.2, étant donné les B_s , on peut générer les arbres qui minimisent $\sum_s B_s$. Ainsi, si on peut mesurer ces probabilités (en comptant le nombre d'appels bloqués sur les arcs), il nous est possible de générer de bons arbres sans connaître les matrices de trafic. Si un appel qui ne correspond pas à un type d'appel pré-défini arrive, nous pouvons générer à l'aide de l'heuristique un arbre qui minimise $\sum_s B_s$ pour le connecter.

L'opérateur du réseau peut alors décider quand il doit mettre à jour les types d'appel. Si on considère que l'appel qui vient d'arriver n'est pas susceptible de se reproduire, on se contente de générer un arbre qui minimise $\sum_s B_s$. Si par contre on constate qu'on a des appels fréquents venant d'une nouvelle source vers un nouvel ensemble de destinations, il faut mettre à jour les types d'appels et ré-optimiser l'acheminement. Si on constate que même en ré-optimisant l'acheminement des appels, la performance du réseau est inacceptable alors, il faudra re-sélectionner les arbres. Si la performance du réseau est toujours inacceptable, l'opérateur n'a pas le choix que de re-dimensionner le réseau afin de tenir compte de l'augmentation du trafic offert au réseau. On doit alors connaître les nouvelles matrices de trafic. En pratique, celles-ci peuvent aussi être mesurées en comptant le nombre de requêtes de connexions venant des commutateurs.

Ainsi, afin de mettre en œuvre un algorithme pour effectuer l'acheminement en temps réel, on a besoin de compteurs pour mesurer les probabilités de blocage des

arcs et le trafic offert au réseau. Une question serait alors s'il y a moyen d'implanter un tel algorithme de façon distribuée plutôt que d'une façon centralisée où les données concernant les valeurs mesurées sont envoyées périodiquement à un serveur qui utilise la méthode heuristique pour générer les arbres.

4.5 Autres méthodes d'optimisation de la sélection d'arbres

Il existe dans la littérature des méthodes d'optimisation autres que celles que nous proposons pour résoudre des problèmes mixtes (à variables entières et variables continues). Parmi ces méthodes, les plus populaires sont actuellement les algorithmes génétiques et les algorithmes de recherche de type tabou. Comme on va le voir dans ce qui suit, le principe sur lequel se basent ces deux méthodes est le même et il consiste à altérer une solution courante afin de voir si on peut l'améliorer. Nous allons expliquer comment notre méthode heuristique repose pratiquement sur le même principe et pourquoi elle est plus adéquate pour résoudre le problème de sélection d'arbres.

4.5.1 Algorithmes génétiques

Les algorithmes génétiques ont été introduits dans [47]. Il s'agit d'heuristiques qui simulent les principes d'évolution dans une population en considérant quatre opérations:

1. sélection,
2. croisement,

3. mutation et

4. évaluation.

Chaque opération est répétée, soit un nombre prédéterminé de fois, soit jusqu'à ce que la solution réponde à un critère donné. En pratique, on associe une probabilité qu'un croisement ou une mutation ait lieu. Le croisement qui contrôle la reproduction se fait en général avec une forte probabilité. La mutation se fait en général avec une faible probabilité. Dans la littérature, il n'y a pas de valeurs sur lesquelles les chercheurs semblent s'entendre mais en général, la probabilité de croisement est supérieur à 0.6 et celle de mutation est de l'ordre de 0.01 [22].

Dans le contexte d'acheminement, l'opération sélection se ferait selon un critère qui minimise la somme des coûts induits ou des probabilités de blocage des arcs d'un arbre donné. Afin de tenter d'améliorer la solution courante, on pourrait envisager un croisement d'un certain nombre d'arbres avec d'autres arbres dont le résultat serait un arbre qui utilise le minimum possible d'arcs en commun avec eux. Une mutation serait de modifier un arbre existant en y remplaçant un arc et l'évaluation serait indiquée par le revenu net de l'arbre. On se rend alors compte qu'à peu de choses près, c'est ce que fait la méthode heuristique que nous proposons dans la section 4.3.2.2.

Il est difficile de quantifier la complexité d'un tel algorithme. Si nous voulons envisager toutes les mutations et les croisements possibles, nous obtenons une complexité trop grande, qui peut être plus grande que celle de la méthode exacte proposée dans la section 4.3.1. En effet, il y a un nombre très élevé de façons de remplacer un arc dans un arbre et de considérer les combinaisons des arbres qui subissent de telles mutations (pour tous les types d'appels). Aussi, pour un arbre donné, il y a un

nombre très grand de façons de trouver des arbres disjoints avec lui. De plus, si nous ne contrôlons pas la taille de la population, nous pouvons rapidement se retrouver avec un très grand nombre d'arbres rendant l'optimisation difficile voire impossible avec les outils informatiques dont nous disposons. Il faudra alors limiter la taille de la population et c'est ce que fait la méthode heuristique que nous proposons dans la section 4.3.2.2 en limitant le nombre de nœuds de Steiner dans les arbres sélectionnés. Il faut aussi noter qu'en pratique, les algorithmes génétiques sont utilisés pour des population de tailles limitées, en général, quelques centaines [29, 22].

4.5.2 Recherche de type tabou

La recherche de type tabou a été introduite dans [42]. Brièvement, il s'agit d'une heuristique qui consiste à partir d'une solution courante et définir un voisinage en altérant légèrement cette solution selon des transformations spécifiques. On obtient ainsi un ensemble de solutions dont la meilleure devient celle à partir de laquelle commence l'étape suivante. Une particularité intéressante de cette méthode est que les transformations peuvent dégrader la fonction objectif ce qui évite la convergence rapide vers de mauvais optimums locaux. Aussi, pour éviter le retour vers des solutions déjà explorées, on interdit les solutions qui ont été explorées récemment en les considérant «tabou», d'où l'appellation de la méthode.

Dans le problème de sélection d'arbres et dans le contexte d'acheminement, nous pouvons définir le voisinage d'une solution comme étant l'ensemble d'arbres qui diffèrent d'un arc de ceux qui écoulent actuellement du trafic. Même avec une aussi petite différence, en pratique, ce voisinage peut être immense. En effet, pour un arbre donné, et dans un réseau de taille N , la taille du voisinage est $O(N^3)$ car il y

a $N - 1$ façons d'enlever un arc d'un arbre et $N(N - 1)/2 - (N - 1)$ façons de le remplacer. Si on a KCM arbres en tout, la taille du voisinage devient $O(N^{3KCM})$. Afin de limiter la taille de ce voisinage, nous pouvons plutôt ne changer un arc qu'avec un arc qui satisfait un certain critère par exemple, dont la probabilité de blocage $B_s(c)$ (ou le coût induit) et au plus égale à la sienne. Nous optimisons alors l'acheminement et nous retenons seulement les arbres écoulant du trafic. Ici aussi, nous constatons qu'à peu de choses près, c'est ce que fait la méthode heuristique que nous proposons dans la section 4.3.2.2.

4.5.3 Résumé

Nous avons brièvement expliqué comment on pourrait sélectionner les arbres en utilisant des méthodes génétiques ou de recherche de type tabou. Nous pouvons conclure qu'il existe des ressemblances entre ces méthodes et notre heuristique mais l'heuristique que nous proposons est plus appropriée pour résoudre le problème de sélection d'arbres car elle a été spécialement développée pour cette fin.

4.6 Conclusion

Nous avons illustré dans quelle mesure la sélection d'arbres avait un impact sur les performances du réseau et nous avons proposé un critère simple pour sélectionner les arbres. Nous avons alors présenté deux algorithmes pour la sélection d'arbres dans le contexte d'acheminement. Le premier algorithme converge théoriquement vers une solution exacte (un optimum local) mais il est très complexe et la qualité de ses solutions dépend considérablement de la solution initiale. Le deuxième algorithme est basé sur une méthode heuristique et nous permet de trouver rapidement une

bonne sélection d'arbres qui peut servir entre autre comme solution initiale pour la méthode exacte. De plus, pour plusieurs exemples, les solutions obtenues par la méthode heuristique sont optimales pour la méthode exacte ce qui veut dire que pour ces exemples, la méthode exacte n'est pas nécessaire pour trouver une bonne solution optimale.

CHAPITRE 5

Sélection d'arbres pour la synthèse de réseaux

Nous donnons dans ce chapitre une description du problème de sélection d'arbres dans le contexte de la synthèse de réseaux. Dans la section 5.1, nous donnons une formulation mathématique pour ce problème. Dans la section 5.2, nous résumons une étude que nous avons effectuée pour étudier l'impact de la sélection d'arbres sur la synthèse de réseaux et nous donnons ensuite un critère simple pour la sélection d'arbre. Dans la section 5.3, nous expliquons comment on pourrait utiliser les méthodes du gradient projeté et de génération de colonnes pour résoudre le problème de sélection d'arbre de façon exacte. Enfin, dans la section 5.4, nous proposons une méthode heuristique simple pour résoudre le problème de sélection d'arbres, qui a fait l'objet de l'article [73]. Les deux méthodes ont fait l'objet de l'article [71].

5.1 Formulation du problème

De même que pour l'acheminement, on peut énoncer le problème de sélection d'arbres dans le contexte de la synthèse de réseaux comme suit:

pour une matrice de trafic donnée, trouver le triplet (S, α, N) incluant l'ensemble d'arbres S , l'acheminement α et les capacités des artères de transmission N qui maximise la valeur nette du réseau.

Ainsi, les problèmes de sélection d'arbres, de dimensionnement et d'acheminement doivent être traités simultanément. La formulation mathématique du problème est

$$\min_{\alpha, N, S} z = \sum_s C_s N_s - \sum_{k, c, m} w^k(c) A^k(c) \alpha_m^k(c) Q_m^k(c) \quad (5.1)$$

$$\sum_{m=0} \alpha_m^k(c) = 1 \quad (5.2)$$

$$\alpha_m^k(c) \geq 0 \quad (5.3)$$

$$L^k(c) \leq \bar{L}^k(c) \quad (5.4)$$

$$N_s \geq 0 \quad \forall s \quad (5.5)$$

où C_s est une constante qui donne le coût d'une unité de capacité, sur l'arc s et $\bar{L}^k(c)$ est la probabilité de perte d'appel maximale permise pour le type k de classe c .

Dans cette formulation, on peut identifier le revenu des connexions par

$$z_r = \sum_{k, c, m} w^k(c) A^k(c) \alpha_m^k(c) Q_m^k(c)$$

et le coût du réseau par

$$z_c = \sum_s C_s N_s.$$

Il est clair que le problème de synthèse est encore plus complexe que le problème d'acheminement puisqu'on optimise aussi les capacités. Cependant, comme on va le voir dans la section 5.2, il existe un critère simple pour sélectionner les arbres de façon à maximiser la valeur nette du réseau. Ce critère nous a permis de développer une heuristique assez simple que nous décrivons dans la section 5.4.

5.2 Critère de sélection d'arbres

Afin de trouver un critère de sélection d'arbres, nous allons évaluer l'impact du choix d'arbres sur la performance du réseau. Dans cette section, nous utilisons les mêmes définitions que dans la section 4.2.

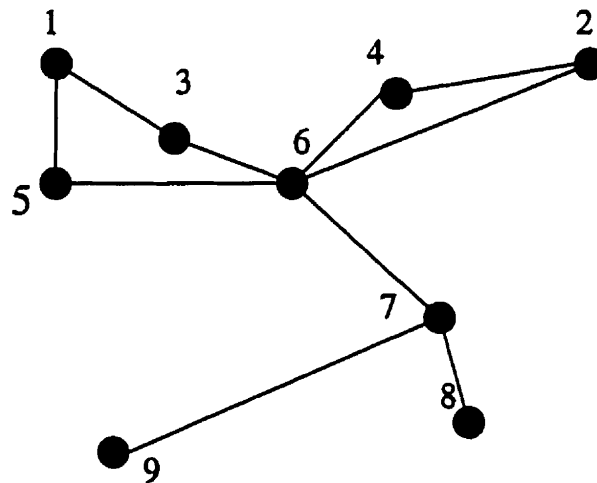
Par ailleurs, nous allons supposer que le coût de capacité est le même pour tous les arcs c.-à-d., $C_s = C_0 \forall s$. Cette hypothèse est réaliste dans la mesure où nous avons des arcs de même type (fibre par exemple) et que la capacité a le même coût pour tous les arcs du réseau.

Exemple 5.2.1 Considérons le réseau donné dans l'exemple 4.2.2 avec la sélection d'arbres de la figure 4.5. Quand nous optimisons le dimensionnement avec l'algorithme de la section 3.4.2, nous avons constaté que tout le trafic de type 1 est écoulé sur l'arbre de Steiner, tout le trafic de type 2 est écoulé sur l'arbre de recouvrement et enfin, le trafic de type 3 est écoulé sur l'arbre de Steiner. Ceci donne lieu à la topologie optimale donnée dans la figure 5.1. Nous avons seulement dix arcs dans le réseau et en supposant que $C_0 = 0.1$, le coût du réseau est 14.9.

D'autre part, nous avons constaté qu'à la solution optimale, les coûts induits de tous les arcs sont du même ordre de grandeur. Ceci est dû au fait que dans la mesure où on peut ajuster les capacités des arcs, on peut réduire leurs coût induit

Tableau 5.1: *QS pour le dimensionnement optimal du réseau.*

k	1	2	3
L^k	0.01	0.01	0.01

Figure 5.1: *Topologie optimale du réseau.*

en augmentant leurs capacités. Lorsqu'on faisait uniquement de l'acheminement, le seul moyen de réduire les coûts induits des arcs était d'acheminer le trafic sur des arbres disjoints et dans ce cas, il n'est pas toujours possible d'éviter les arcs coûteux.

Dans la mesure où dans l'exemple 4.2.2 nous avons 23 arcs de capacité $N_s = 10$ chacun, le coût du réseau était de 23.0. Cela illustre dans quelle mesure la concentration du trafic réduit le coût du réseau. Le revenu des appels est 69.3 et la QS est donnée dans le tableau 5.1. Ainsi, on peut dire qu'en satisfaisant les contraintes de QS, nous avons réussi à réduire le coût du réseau de 35%. Ce résultat est très important car il montre à quel point nous pouvons faire des économies en concentrant le trafic.

Maintenant, nous allons sélectionner les arbres de façon à ce que ceux-ci aient le maximum possible d'arcs en commun. Nous utilisons alors les arbres de la figure 5.2. Comme on peut voir sur cette figure, les arbres d'indice 2 utilisent huit arcs en tout. Les arbres d'indice 1 sont des étoiles que nous utilisons pour illustrer la non convexité du problème de synthèse de réseaux.

Quand nous optimisons le dimensionnement avec l'algorithme de la section 3.4.2, nous trouvons que tous les appels sont connectés sur les arbres d'indice 2. Nous obtenons alors un réseau de huit arcs (un arbre) dont le coût est 13. On a pratiquement le même revenu et QS que le cas où nous utilisons la sélection d'arbres de la figure 4.5. Comme le coût du réseau avec cette sélection d'arbres était de 14.9, nous avons fait des économies 13% en utilisant des arbres ayant plusieurs arcs en commun.

Quand nous commençons par une solution initiale qui privilégie les étoiles (de forte valeurs de α_m^k pour les arbres correspondant), nous obtenons un réseau où tout le trafic est écoulé par les étoiles. Ce réseau a 15 arcs et son coût est de 16.3 qui est 20% plus cher que le réseau de huit arcs. Ainsi, nous illustrons à la fois la non convexité du problème et l'inconvénient de disperser le trafic sur plusieurs arcs dans le contexte de la synthèse de réseaux. Ces résultats sont résumés dans le tableau 5.2.

Exemple 5.2.2 Dans l'exemple 4.2.3, nous avons utilisé 30 arcs de capacité 17 chacun ce qui veut dire que le coût du réseau est 51 et ce pour $C_0 = 0.1$.

Nous allons maintenant optimiser les capacités des arcs pour les données et le choix d'arbres de l'exemple 4.2.3. Pour ce faire, nous utilisons l'algorithme de dimensionnement de la section 3.4.2.

En dimensionnant les arcs, nous constatons que comme dans l'exemple 5.2.1, un

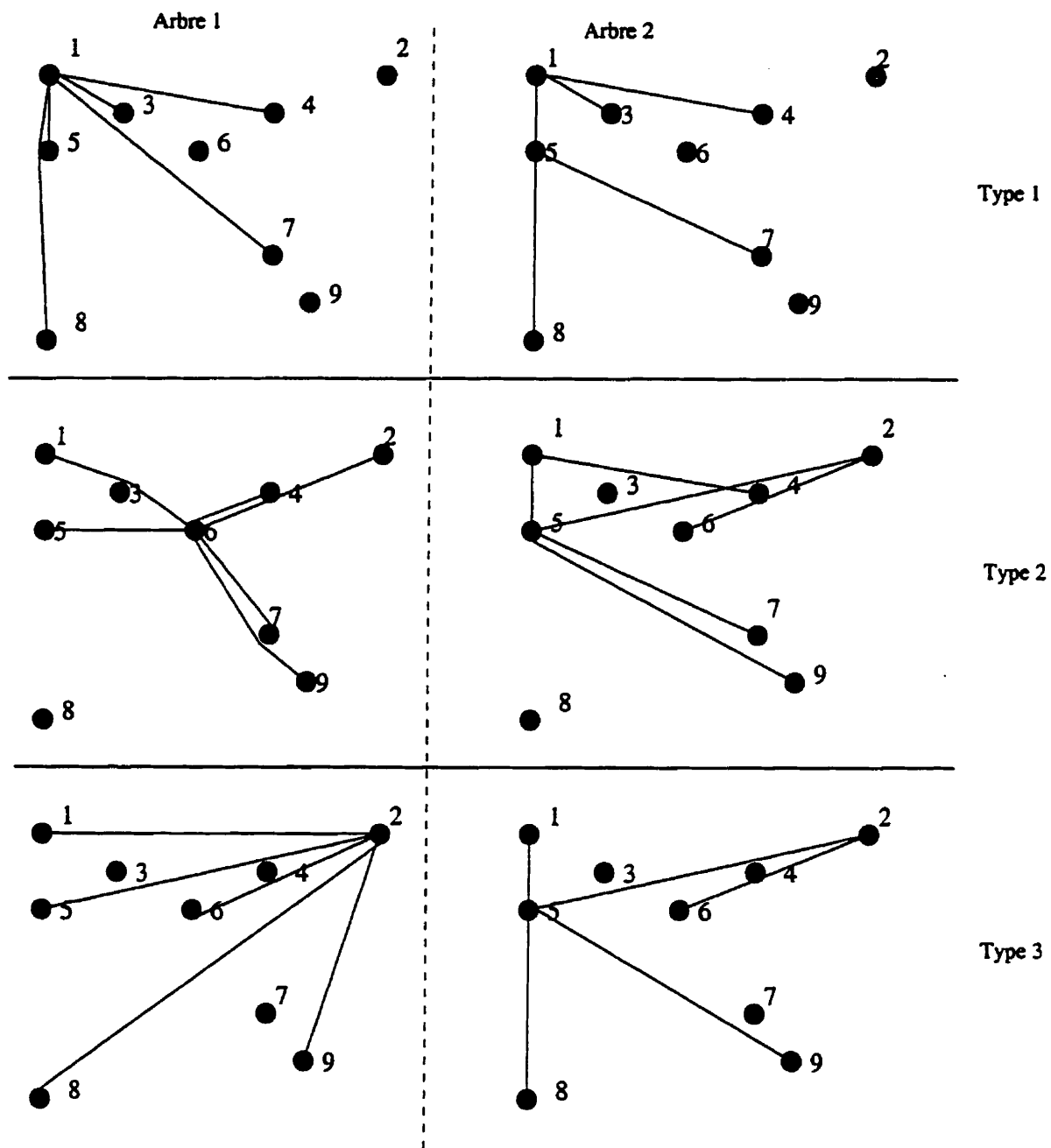


Figure 5.2: Sélection d'arbres ayant plusieurs arcs en commun.

Tableau 5.2: *Influence des arcs communs sur la synthèse de réseaux.*

	Arcs				
Sélection d'arbres	Utilisés	Communs	Coût	Revenu	$L^1; L^2; L^3$ (%)
8 arcs en commun	8	7	13.04	69.32	1; 1; 0.88
étoile, MST, Steiner	10	6	14.92	69.30	1; 1; 1.
étoiles	15	1	16.30	69.30	1; 1; 1

seul arbre est utilisé pour écouler les appels d'un type d'appel donné. Ceci correspond à l'arbre de recouvrement pour le type 1 de classe 1, l'arbre de recouvrement pour le type 2 de classe 1, l'arbre de Steiner pour le type 3 de classe 1, l'arbre de recouvrement pour le type 1 de classe 2 et enfin, l'arbre de recouvrement pour le type 2 de classe 2. Comme nous pouvons le voir dans la figure 4.7, ces arbres ont plusieurs arcs en commun. Nous obtenons alors le réseau de 15 arcs de la figure 5.3 dont le coût est 29.7 ce qui veut dire que nous avons réalisé des économies de 42% par rapport au réseau utilisé dans l'exemple 4.2.3. Le revenu est 132.7 et la QS est donnée dans le tableau 5.3.

Nous pouvons encore réduire le coût du réseau en concentrant davantage le trafic. Il est clair que dès que le nombre d'appels devient important, le choix des arbres devient difficile à faire. Cela illustre le besoin de développer une approche systématique pour sélectionner les arbres. En effet, comme on le verra dans l'exemple 5.4.2, on peut réduire le coût du réseau considérablement en concentrant le trafic sur des arcs bien choisis.

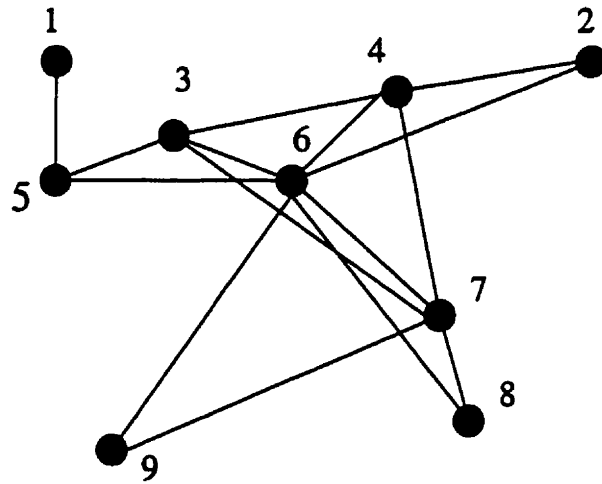


Figure 5.3: *Topologie optimale du réseau avec multi-débit.*

Tableau 5.3: *QS pour le dimensionnement optimal du réseau avec du multi-débit.*

(k, c)	(1, 1)	(2, 1)	(3, 1)	(1, 2)	(2, 2)
$L^k(c)$	0.011	0.010	0.009	0.010	0.010

5.2.1 Influence du facteur de coût C_0 et des revenus $w^k(c)$

Afin de tester l'influence du paramètre C_0 ou des revenus $w^k(c)$ sur les résultats, nous avons dimensionné le réseau avec la même sélection d'arbres que dans l'exemple 5.2.1, pour différentes valeurs de C_0 . Les résultats sont donnés dans le tableau 5.4 pour l'acheminement et la QS et dans le tableau 5.5 pour les capacités des arcs.

D'après ces résultats, nous constatons que C_0 n'affecte pas le choix des arbres. Seul le coût du réseau augmente ou diminue proportionnellement à C_0 . Cependant, nous avons constaté que les capacités installées sur les arcs correspondant aux arbres sélectionnés augmentaient légèrement si C_0 diminue considérablement et ceci est tout à fait prévisible puisque la capacité coûte moins cher. Néanmoins, il faut une très forte variation de C_0 (dix fois) pour observer une variation significative des capacités. Aussi, à partir d'une certaine valeur de C_0 , les capacités optimales ne changent pratiquement plus. Cette limite correspond aux capacités minimales requises pour satisfaire les contraintes de QS.

D'autre part, comme nous pouvons le constater dans le tableau 5.4, plus C_0 augmente, plus le coût rattaché à la perte $L^k(c)x^k(c)$ est important et plus les revenus nets marginaux pondérés des arbres sont faibles. Ceci est tout à fait prévisible puisque plus la capacité coûte cher, plus il est coûteux (ou encore, moins cela génère du revenu) d'écouler un appel supplémentaire dans le réseau.

Enfin, il est important de noter que le choix de C_0 est intimement lié aux revenus w^k . En pratique, pour des revenus valant 1 par destination, nous avons constaté que pour une plage de valeurs $0.01 \leq C_0 \leq 0.5$, nous obtenons pratiquement le même réseau. Si nous augmentons les w^k et C_0 proportionnellement, cela ne change

Tableau 5.4: *Acheminement et QS pour le dimensionnement du réseau vs C_0 .*

C_0	Coût	Revenu	k	1	2	3
0.01	1.7	69.9	L^k	0.01	0.01	0.01
			$L^k x^k$	0.	0.	0.
			m	3	2	3
			\overline{R}_m^k	19.59	29.49	17.77
0.1	14.9	69.3	L^k	0.01	0.01	0.01
			$L^k x^k$	0.20	0.07	0.07
			m	3	2	3
			\overline{R}_m^k	16.63	25.99	18.19
0.2	29.8	69.3	L^k	0.01	0.01	0.01
			$L^k x^k$	0.60	0.45	0.36
			m	3	2	3
			\overline{R}_m^k	13.46	22.19	16.58
0.5	74.6	69.3	L^k	0.01	0.01	0.01
			$L^k x^k$	1.79	1.54	1.21
			m	3	2	3
			\overline{R}_m^k	4.13	11.02	11.74

Tableau 5.5: Capacités (N_s) des arcs et coût du réseau vs C_0 .

s	(i, j)	$C_0 = 0.01$	$C_0 = 0.1$	$C_0 = 0.2$	$C_0 = 0.5$
2	(1, 3)	12.4	10.8	10.8	10.8
4	(1, 5)	24.5	21.8	21.8	21.8
10	(2, 4)	14.4	12.3	12.2	12.2
12	(2, 6)	8.6	7.2	7.2	7.2
18	(3, 6)	12.4	10.8	10.8	10.8
23	(4, 6)	21.2	18.7	18.7	18.7
27	(5, 6)	18.1	15.7	15.7	15.7
31	(6, 7)	24.5	21.8	21.8	21.8
34	(7, 8)	16.3	14.3	14.3	14.3
35	(7, 9)	18.1	15.8	15.7	15.7
	$C_0 \sum_s N_s$	1.7	14.9	29.8	74.6

absolument rien aux résultats. Ainsi, il suffit de fixer l'un et de faire varier l'autre pour étudier et l'influence de C_0 et celle de w^k . Notons aussi qu'il est raisonnable de faire en sorte que le coût du réseau ne dépasse pas le revenu généré par les appels.

5.2.2 Interprétation des résultats

D'après ces résultats (qui ont été vérifiés sur plusieurs exemples), nous pouvons dégager les points suivants:

- Le problème de synthèse de réseaux n'est pas convexe car il a été possible d'obtenir des solutions optimales différentes pour les mêmes données, mais en partant de solutions initiales différentes. On obtient donc des optimums locaux, ce qui veut dire que les algorithmes que nous proposons ici ne garantissent pas la convergence vers un optimum global.
- Le dimensionnement optimal élimine l'existence d'arcs coûteux car on optimise les capacités des arcs. De ce fait, on élimine les problèmes de perte de revenu due aux arcs coûteux qu'on retrouve dans le cas où on effectue seulement l'acheminement.

On peut aussi dégager un critère simple pour la sélection d'arbres qui s'énonce comme suit:

Dans le contexte de la synthèse de réseaux, il est préférable de concentrer le trafic et d'utiliser des arbres avec le maximum possible d'arcs en commun. Cela donne un réseau optimal faiblement connecté (se réduisant souvent à un arbre). Il vaut donc mieux avoir un petit nombre d'arcs de

grandes capacités plutôt qu'un grand nombre d'arcs de capacités relativement faibles. De plus, le dimensionnement optimal conduit à l'utilisation d'un arbre unique pour connecter les appels point-à-multipoint d'un type donné.

Ainsi, on obtient un critère de sélection contraire à celui obtenu dans le contexte d'acheminement (voir section 4.2). Ceci est essentiellement dû au fait que pour l'acheminement, on ne peut diminuer les coûts induits des arcs qu'en acheminant les appels sur des arbres disjoints. Cela conduit à l'utilisation de plusieurs arbres par type d'appel donc, à l'utilisation de plusieurs arcs. Par contre, si on peut ajuster les capacités des arcs, le problème d'arcs coûteux est éliminé et il est plus rentable de faire en sorte que les arbres des différents types d'appels partagent le maximum possible d'arcs donnant lieu à une topologie optimale faiblement connectée [70]. Il faut noter que ceci repose sur l'hypothèse de la concavité de la fonction qui donne le coût des capacités des arcs ce qui est le cas de notre modèle où ces coûts sont linéaires. Dans le cas de fonction coût non concave, on ne sait pas quelle serait la topologie optimale du réseau. Cependant, il faut noter qu'en pratique, les coûts sont concaves dans la mesure où il est moins cher d'acheter un petit nombre de systèmes de grosses capacités que plusieurs systèmes de capacités relativement petites.

5.2.3 Résumé

Nous avons illustré dans quelle mesure la concentration de trafic permet de réduire le coût du réseau. Il faut maintenant trouver une approche systématique pour générer et sélectionner les arbres dans le contexte de la synthèse de réseaux. Nous proposons deux méthodes, une méthode exacte qui est malheureusement inutilisable

en pratique et une méthode heuristique basée sur le critère de sélection qu'on vient de décrire.

5.3 Méthode exacte

Si on connaissait *tous* les arbres candidats correspondant à tous les types d'appel et qu'on optimisait le dimensionnement et l'acheminement, seuls les arbres écouant du flot positif auraient dû être sélectionnés pour connecter les appels et nous aurions alors résolu le problème de sélection d'arbres. Malheureusement, il est pratiquement impossible de connaître tous les arbres pouvant être utilisés pour connecter les appels. Nous avons alors besoin d'une méthode qui ne requiert pas la connaissance à priori de ces arbres. Ceci peut être réalisé à l'aide des méthodes du gradient projeté [67] et de génération de colonnes.

5.3.1 Gradient projeté

Dans la section 4.3.1, nous avons montré que la méthode du gradient projeté peut être utilisée pour résoudre le problème de sélection d'arbres dans le contexte d'acheminement d'appels. Nous pouvons utiliser exactement la même approche ici sauf que nous avons comme variables les α et les N aussi. Nous pouvons facilement montrer que d'une itération à l'autre du gradient projeté, on a

$$\alpha_m^k(c) \leftarrow \alpha_m^k(c) - \beta \frac{\partial z}{\partial \alpha_m^k(c)} + \frac{\beta}{n} \sum_{i \notin \Omega_c^k(\alpha)} \frac{\partial z}{\partial \alpha_i^k(c)} \quad \forall m \notin \Omega_c^k(\alpha)$$

et

$$N_s \leftarrow N_s - \beta \frac{\partial z}{\partial N_s} \quad \forall s \notin \Omega(N)$$

où $\Omega_c^k(\alpha)$ et $\Omega(N)$ sont les ensembles de contraintes actives correspondant aux coefficients de partage de charge et aux capacités des arcs, respectivement.

Cet algorithme converge localement dans l'espace défini par les contraintes actives et les conditions de convergence sont exactement celles données dans la section 4.3.1. Une fois que nous avons cet optimum local, nous voudrions alors savoir quel(s) arbre(s) et/ou arc(s) devrions nous faire sortir de l'ensemble des contraintes actives afin d'améliorer la fonction objectif; ce qui peut être réalisé par la méthode de génération de colonnes.

5.3.2 Génération de colonnes

Dans le contexte d'acheminement, la génération de colonne concernait seulement l'ensemble $\Omega(\alpha)$. Ici, elle concerne aussi l'ensemble de contraintes actives $\Omega(N)$ rendant ainsi le problème de génération d'arbres encore plus complexe.

Dans la section 4.3.1, nous avons montré que dans le contexte d'acheminement, dans le cas où nous trouvons un arbre satisfaisant $\bar{R}_m^k(c) > v^k(c)$ ($u_m^k < 0$), cet arbre quitte l'ensemble des contraintes actives puisqu'il génère plus de revenu que les arbres écoulant du flot actuellement. Nous pouvons généraliser ceci au contexte de la synthèse de réseaux excepté que nous devons considérer $\bar{\bar{R}}_m^k(c)$ au lieu de $\bar{R}_m^k(c)$. En effet, comme nous le voyons dans les équations (3.30) et (3.31), les arbres écoulant du flot positif ont des revenus $\bar{\bar{R}}_m^k(c)$ égaux et les arbres n'écoulant pas de flot, ont un revenu inférieur. Ainsi, en maximisant $\bar{\bar{R}}_m^k(c)$, si nous pouvons générer un arbre qui satisfait $u_m^k(c) < 0$, cet arbre est celui qui doit quitter l'ensemble des contraintes actives $\Omega_c^k(\alpha)$.

Malheureusement, il est pratiquement impossible de générer des arbres qui maximisent $\overline{R}_m^k(c)$ car, comme nous l'indique l'équation (3.29), nous devons connaître $L^k(c)x^k(c)$ afin de générer de tels arbres. De plus, nous avons besoin de connaître $Q_m^k(c)$ qui contient le produit des probabilités de connexion sur les arcs alors que tous les algorithmes de génération d'arbres que nous connaissons sont basés sur des coûts additifs. Pour pallier cette difficulté, dans la section 4.3.1, nous avons supposé que les $Q_m^k(c)$ sont proches de 1, et de cette façon, nous avons pu les négliger en générant les arbres. Or, ici nous avons le produit $L^k(c)x^k(c)$ dans $\overline{R}_m^k(c)$ et en pratique, il est facile de montrer avec des résultats numériques que ce produit ne peut être négligé devant $\overline{R}_m^k(c)$, même si $L^k(c) \ll 1$ (car $Q_m^k(c) \cong 1$). Un exemple est donné dans le tableau 5.4 pour $C_0 = 0.5$. Notons aussi que les $L^k(c)$ sont une fonction du produit des probabilités de blocage des arcs ce qui rend la résolution du problème d'arbre de Steiner très difficile voire, impossible.

De plus, on a besoin de connaître quelles contraintes (5.5) doivent quitter $\Omega(N)$. L'ensemble de contraintes (5.5) inactives va ainsi déterminer les arcs que nous pouvons considérer lors de la génération d'arbres tel que décrit ci-dessus.

Afin de connaître quelle contrainte (5.5) doit quitter $\Omega(N)$ (et l'arc s correspondant), nous avons besoin de chercher s'il existe un arbre (k, c, m') qui utilise l'arc correspondant à cette contrainte ($\mathcal{I}_{s,(k,c,m')} = 1$) et qui satisfait

$$\overline{R}_{m'}^k(c) > \overline{R}_m^k(c) \quad \forall m \notin \Omega_c^k(\alpha).$$

Ainsi, les ensembles $\Omega(N)$ et $\Omega_c^k(\alpha)$ sont intimement liés ce qui rend le problème de sélection d'arbres encore plus complexe. Par exemple, lorsque nous sommes entrain de dimensionner le réseau, quelques variables N_s peuvent s'annuler, ce qui va rendre les contraintes qui leur correspondent actives. Ainsi, toutes les contraintes (5.3)

correspondant aux arbres utilisant les arcs pour lesquels $N_s = 0$ deviennent aussi actives. Cela rend la résolution du problème de sélection d'arbres de façon exacte pratiquement impossible. C'est pourquoi, nous ne pouvons qu'utiliser des heuristiques pour résoudre ce problème.

5.4 Méthode heuristique

Dans cette section, nous proposons des approximations afin de simplifier le problème de sélection d'arbres et nous allons illustrer pourquoi la concentration de trafic permet de réduire le coût du réseau. Pour le revenu, nous allons utiliser les approximations de la section 4.3.2.1. Notons qu'ici, une approximation notée par \tilde{X} pour une variable X donnée ne correspond pas nécessairement à une borne inférieure ou supérieure.

5.4.1 Approximation du coût et des contraintes de QS

En supposant que la topologie du réseau et les contraintes de QS sont données, nous pouvons estimer les capacités qui doivent être installées sur les arcs. Nous allons remplacer à cette fin les contraintes de QS (5.4) de bout en bout par des contraintes sur les probabilités de blocages des arcs.

Pour un arc s donné, soit la classe c^* satisfaisant $W(c^*) \geq W(c) \forall c$ sur cet arc. Ainsi, de l'équation (4.15), et en supposant que $L_m^k(c^*) = \bar{L}^k(c^*)$ pour tous les arbres écoulant du flot, nous pouvons déduire que

$$\tilde{B}_s(c^*) = \min_{\{(k,m)\} | s \in (k,c^*,m)} \frac{\bar{L}^k(c^*)}{D^k(c^*)} \quad (5.6)$$

où $D^k(c^*)$ est le nombre de destinations pour le type d'appel k de classe c^* . Dans

l'équation (5.6), nous avons supposé que nous avons un arbre dont le nombre d'arcs est $D^k(c^*)$ ce qui est le cas des arbres de recouvrement. En pratique, nous pouvons avoir des arbres de Steiner mais nous savons qu'il n'est pas économique d'utiliser de très gros arbres dans la mesure où ceux-ci génèrent de faibles revenus [40, 70]. Nous pouvons alors déduire l'approximation pour la capacité de l'arc s

$$\tilde{N}_s = g_c^{-1}(\tilde{B}_s(c^*), \mathbf{W}, \hat{\mathbf{a}}_s). \quad (5.7)$$

où $\hat{\mathbf{a}}_s$ est évalué par l'équation (4.11). L'équation (5.7) stipule que la capacité de l'arc s doit être celle qui fait en sorte que les appels de classe c^* aient une probabilité de blocage $\tilde{B}_s(c^*)$ et puisque pour toutes les autres classes nous avons $\tilde{B}_s(c) \leq \tilde{B}_s(c^*)$, alors, les contraintes de QS seront satisfaites pour tous les types d'appel si l'équation (5.7) est vraie pour tous les arcs. Ceci est vrai car nous n'avons pas de réservation de circuit dans notre modèle. Si cela était le cas, il faudra trouver une autre démarche pour approximer les capacités.

Enfin, nous pouvons approximer le coût du réseau par

$$\tilde{z}_c = C_0 \sum_s \tilde{N}_s \quad (5.8)$$

et la valeur nette du réseau par

$$\tilde{z} = \tilde{z}_c - \hat{z}_r \quad (5.9)$$

qui peuvent être facilement évalués connaissant les $\bar{L}^k(c)$ et les arbres.

5.4.2 Concentration de trafic

Pour le problème de synthèse de réseaux, nous avons constaté sur tous les exemples numériques que nous avons considérés (une cinquantaine) que pour chaque type d'appel, un arbre unique est utilisé pour acheminer le trafic et que tous les arbres

sélectionnés utilisent plusieurs arcs en commun. Nous avons aussi constaté que plus nous réduisons le nombre d'arcs dans le réseau, moins ce dernier coûte cher. Ainsi, intuitivement, on devrait s'attendre à ce que le réseau qui coûte le moins cher soit un réseau très faiblement connecté se réduisant souvent à un arbre. Ce résultat n'est pas surprenant dans la mesure où il est connu que les systèmes avec grosses capacités ont de meilleures performances marginales que les systèmes à faibles capacités. Nous pouvons expliquer ce phénomène comme suit.

Dans le cas d'une seule classe d'appel, nous savons empiriquement que la fonction f donnée par l'équation (5.10) (où nous avons supprimé l'indice de classe) est concave. D'autres propriétés intéressantes de cette fonction sont que $f(0) = 0$ et qu'elle est asymptotiquement linéaire (voir figure 5.4 où nous donnons cette fonction pour $B_s = Ct = 0.001$).

$$N_s = f(a_s) |_{B_s=Ct}. \quad (5.10)$$

Ainsi, nous pouvons faire la proposition « de concentration de trafic » suivante:

Proposition 5.4.1 *Nous supposons que la fonction f donnant la capacité d'un arc s à blocage constant sur cet arc est concave et strictement croissante. Alors, la capacité marginale ΔN_s , requise pour augmenter le trafic offert à l'arc s d'une quantité $\Delta a_s > 0$ à blocage constant sur cet arc, diminue à mesure que la capacité de l'arc augmente.*

Démonstration: Nous allons faire une démonstration graphique (une démonstration analytique est donnée dans l'annexe A). Considérons la fonction f donnée dans la figure 5.4. Il est clair que pour maintenir la probabilité de blocage constante sur un arc s donné, accroître le trafic par une quantité marginale Δa_s requiert une plus petite capacité marginale dans le cas où nous avons une grosse capacité sur l'arc. En d'autres mots, $N'_s < N_s \Rightarrow \Delta N'_s > \Delta N_s$.

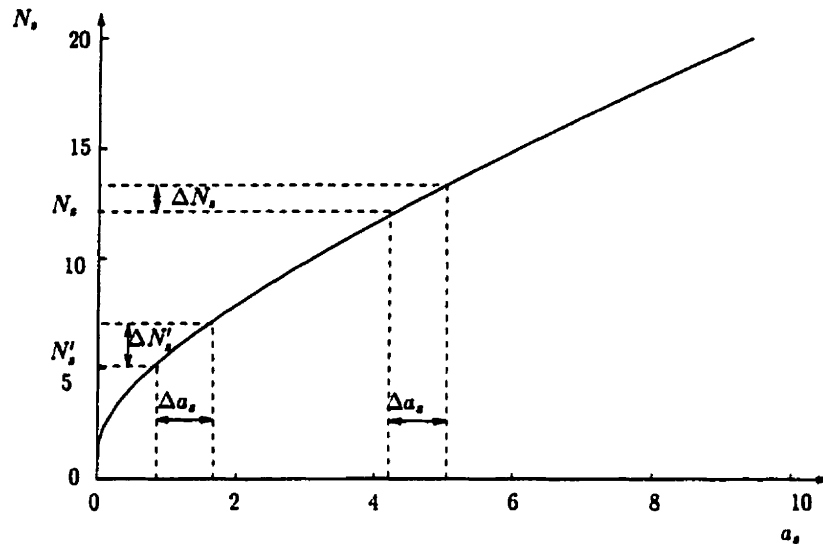


Figure 5.4: Variation de la capacité en fonction du trafic offert à blocage constant ($B_s = 0.001$).

Ainsi, il est économique de concentrer le trafic sur un petit nombre d'arcs de grosses capacités plutôt que sur un grand nombre d'arcs de capacités relativement petites. D'autre part, nous pouvons faire la proposition « du chemin unique » suivante :

Proposition 5.4.2 *Nous dimensionnons un réseau de N nœuds ($N \geq 3$) à blocage constant et égal à B_0 sur tous les arcs du réseau (mono-débit). Nous supposons que la fonction f donnant la capacité d'un arcs s à blocage constant sur cet arc est concave et strictement croissante et que $f(0) = 0$. Nous voulons acheminer une quantité de trafic a (trafic offert) d'un nœud A vers un nœud B donnés. Nous supposons que nous n'avons pas encore installé de capacité entre A et B et que nous allons utiliser des arcs qui n'écoulent pas de trafic actuellement. Dans ces conditions, il suffit que $B_0 \leq 1/(N - 1)$ pour qu'il soit plus économique d'écouler le trafic de A vers B sur un chemin unique, ayant le plus petit nombre possible d'arcs.*

Les nœuds A et B peuvent être la source et la destination d'un type d'appel point-à-point, des nœuds de transit appartenant à une communication point-à-point ou des nœuds figurant sur un arbre connectant des appels point-à-multipoint. En général, le trafic écoulé de A vers B est une agrégation de trafics provenant de plusieurs types d'appel.

Une démonstration rigoureuse de cette proposition est donnée dans l'annexe B. Pour avoir une idée sur notre démarche, considérons la figure 5.5 où nous avons deux cas: un cas où le trafic a est acheminé sur un seul chemin de A vers B et un cas où on utilise deux chemins. En supposant que dans chacun des deux cas les probabilités de blocage sur les arcs sont très faibles ($B_s(c) \ll 1$) et ainsi, que le trafic offert aux arcs est à égal au trafic écoulé a (voir section 4.3.2.1), pour montrer que l'on a besoin de moins de capacité dans le cas où on utilise un seul chemin, il suffit de montrer pour $0 \leq \alpha \leq 1$ que

$$f(a) \leq f(\alpha a) + f((1 - \alpha)a) \quad (5.11)$$

ce qui peut être facilement déduit des propriétés de la fonction f (voir lemme B.1.1).

Dans la figure 5.5, nous avons considéré deux chemins utilisant le même nombre d'arcs. Si cela n'est pas le cas, il est évident que les appels doivent être acheminés sur le chemin ayant le plus petit nombre d'arcs.

Notons que la condition suffisante

$$B_0 \leq \frac{1}{(N - 1)} \quad (5.12)$$

de la proposition 5.4.2 est pratiquement toujours satisfaite pour des réseaux ayant des contraintes de QS raisonnables ($\bar{L}^k(c)$ de l'ordre de 1%). En effet, $B_s(c) \leq \bar{L}^k(c)$

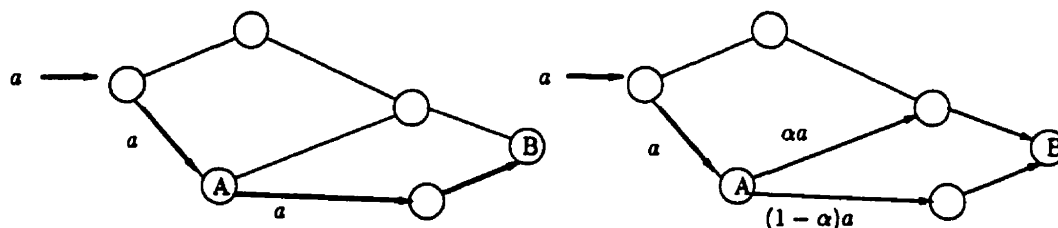


Figure 5.5: *Concentration de trafic sur un seul chemin.*

car autrement, on aura certainement plus que $\bar{L}^k(c)$ comme probabilité de perte d'appel. Par exemple, si $\bar{L}^k(c) = 0.01$, tant que $N \leq 100$, la condition (5.12) est toujours satisfaite. Notons aussi qu'en pratique, $B_s(c)$ est approximée par l'équation (5.6) ce qui veut dire que si le nombre de nœuds dans le réseau (N) augmente, le nombre de destinations ($D^k(c)$) est susceptible d'augmenter aussi faisant en sorte que la condition (5.12) soit satisfaite.

Ainsi, selon la proposition 5.4.2, si nous n'avons pas encore installé de capacité sur des arcs reliant une paire de nœuds donnée, il est plus économique d'utiliser un seul chemin ayant le plus petit nombre d'arcs possible pour écouler le trafic entre ces nœuds. D'autre part, selon la proposition 5.4.1, si nous avons déjà installé de la capacité sur des arcs reliant cette paire de nœuds, il est plus économique d'augmenter la capacité des arcs qui ont les plus grosses capacités pour écouler une quantité de trafic supplémentaire entre ces nœuds. Ainsi, nous pouvons résumer que la concentration de trafic couvre deux aspects notamment

1. il faut avoir le moins d'arcs possible dans le réseau et
2. les arcs doivent être en mesure d'écouler le plus de trafic possible.

Notons que pour montrer les économies réalisées par concentration de trafic, nous n'avons pas tenu compte des coûts des capacités ($C_s(N_s)$). Tant que ces coûts sont

des fonctions concaves et croissantes de la capacité (ce qui est le cas en pratique), la concentration de trafic est économique.

La concentration de trafic est aussi économique dans le contexte du multi-débit puisque nous savons que dans ce cas, le comportement asymptotique des appels est comme si ces derniers saisissent et libèrent les circuits de façon indépendante [64].

5.4.3 Algorithme de sélection d'arbres

Dans la section 5.3, nous avons montré comment on pourrait résoudre le problème de sélection d'arbres de façon exacte et nous avons aussi expliqué pourquoi il est pratiquement impossible de le faire. Néanmoins, nous pouvons simplifier l'algorithme de sélection d'arbres en supposant qu'à la solution initiale $\Omega(N) = \emptyset$ et qu'en cours d'optimisation, la contrainte (5.5) qui devient active n'est plus reconsidérée ce qui veut dire que si un arc voit sa capacité s'annuler en cours d'optimisation, on ne le reconsidérera plus dans la suite. Ainsi, nous pouvons réduire la génération de colonnes en la considérant uniquement pour les contraintes (5.3) c.-à-d., seulement pour $\Omega(\alpha)$. Compte tenu des économies réalisées avec la concentration de trafic sur un petit nombre d'arcs, il est alors évident qu'un tel algorithme va converger vers une solution où le réseau optimal a un petit nombre d'arcs, souvent vers un réseau en arbre. Il reste alors à savoir quels arcs sélectionner pour former cet arbre.

L'algorithme que nous allons proposer ici est basé sur le principe de concentration de trafic décrit dans la section 5.4.2. D'un côté, nous voulons utiliser le moins d'arcs que possible (aspect 1 de la concentration de trafic) et de l'autre, nous voulons faire en sorte que les arcs puissent écouler le plus de trafic que possible (aspect 2 de la concentration de trafic).

La procédure de sélection d'arbres comporte deux étapes: d'abord on effectue une concentration globale du trafic en considérant tous les types d'appel en même temps et ensuite, nous effectuons une concentration individuelle, en considérant les types d'appel un par un. L'idée principale consiste à mettre de la capacité sur le plus petit nombre possible d'arcs afin de minimiser le coût du réseau. Dans la concentration globale, nous essayons de faire partager les arcs par le plus grand nombre possible de types d'appel, ensuite dans la concentration individuelle, nous essayons de faire en sorte que chaque type d'appel utilise le plus petit nombre possible d'arcs.

5.4.4 Algorithme de concentration globale de trafic

L'algorithme qu'on propose ici concentre le trafic conduisant à un réseau en arbre. Les arbres utilisés pour connecter les appels des différents types sont donc des sous arbres du graphe du réseau. La procédure est la suivante:

1. Pour chaque type k de classe c , identifier $\mathcal{E}^k(c)$ l'ensemble des nœuds participant à ce type d'appel (source et destinations).
2. Pour chaque type k , construire les paires ordonnées de nœuds (i, j) , $i < j \forall (i, j) \in \mathcal{E}^k(c) \times \mathcal{E}^k(c)$. Les arcs correspondant à ces paires forment ainsi les arcs potentiels qui pourraient être utilisés pour connecter les appels de type k .
3. Construire la matrice d'incidence

$$\mathcal{I}_{s,k,c} = \begin{cases} 1 & \text{si l'arc } s \text{ peut être utilisé par le type } k \text{ de classe } c \\ 0 & \text{sinon.} \end{cases} \quad (5.13)$$

4. Pour chaque arc s , calculer

$$\ell_s = \sum_k W(c) A^k(c) \mathcal{J}_{s,k,c},$$

le trafic maximal qui peut être offert à cet arc (sans perte).

5. En utilisant ℓ_s comme longueur d'arcs, construire un arbre de recouvrement de coût maximal.
6. Pour chaque type, construire le sous-arbre qui connecte les nœuds participants aux appels de ce type.

Ainsi, cette procédure permet de concentrer le trafic d'une façon *globale* c.-à-d., en considérant tous les types d'appel. On obtient un réseau en arbre quasi-optimal (RAQO) sur lequel sont générés les arbres des différents types d'appel point-à-multipoint. Pour chaque type d'appel, on a donc un arbre unique pour connecter les appels. On peut montrer moyennant les explications données dans la section 5.4.2 la quasi optimalité d'une telle procédure de sélection d'arbres.

La raison pour laquelle nous considérons que le réseau est optimal est que si nous devons chercher un réseau en arbre, la concentration globale de trafic donne le meilleur en terme de valeur nette du réseau. En général, le réseau optimal n'est pas un arbre (voir section 5.4.5) mais avec exclusivement des appels point-à-multipoint, nous pouvons montrer que souvent, nous avons une solution optimale. Pour ce faire, nous avons utilisé des méthodes de recherche de type tabou que l'on décrit dans la section 5.8.

Dans la mesure où le réseau est un arbre, en cas de panne, il n'y a pas de sécurité au niveau commuté. En général, il est nécessaire d'introduire une certaine sécurité au niveau commuté (point-à-point) mais nous ne savons pas si cela peut se généraliser

pour le cas où nous avons des appels point-à-multipoint. Néanmoins, dans la mesure où nous considérons un réseau composé de liens logiques tels que les VP (*Virtual Path*) en ATM, en cas de panne, nous pouvons re-router les circuits sur d'autres chemins physiques.

Enfin, il est aussi important de noter que puisqu'en général il y a plus d'un arbre de recouvrement de coût maximum, l'algorithme de concentration globale de trafic peut avoir plusieurs solutions. Ainsi, nous avons une flexibilité quant au choix de la solution. La qs peut alors être un critère de choix, par exemple, nous pouvons limiter le degré de chaque nœud du réseau afin de réduire le nombre de copies effectuées à chaque commutateur [55] ou encore, le nombre d'arcs dans les branches de l'arbre afin de minimiser le délai de livraison des unités de données de la source vers les destinations [88].

Exemple 5.4.1 Pour les mêmes données de l'exemple 5.2.1, à l'aide de l'algorithme que nous venons de présenter, nous optimisons le dimensionnement, l'acheminement ainsi que la sélection d'arbres. Ceci nous donne le réseau en arbre quasi optimal (RAQO) de la figure 5.6 dont le coût est cette fois 13.0. Le revenu généré par les appels est 69.3 et $L^k(c) = 0.01 \quad \forall k, c$. Ainsi, à l'aide de l'algorithme de concentration globale de trafic, nous avons encore réduit le coût du réseau de 13% par rapport à l'exemple 5.2.1. Les résultats sont résumés dans le tableau 5.6.

Les capacités et les variables d'arcs avec les approximations sont données dans le tableau 5.7. On peut alors remarquer que les approximations sont très bonnes dans ce cas.

Exemple 5.4.2 Nous optimisons maintenant la sélection d'arbres et le dimensionnement du réseau, pour les données de l'exemple 5.2.2, à l'aide de l'algorithme de

Tableau 5.6: *Synthèse de réseaux avec du point-à-multipoint.*

	Coût	Revenu	Valeur nette	$\max_k L^k$
Réf. (étoile,MST,Steiner)	14.9	62.3	54.4	.010
Concentration de trafic	13.0	62.3	56.3	.010

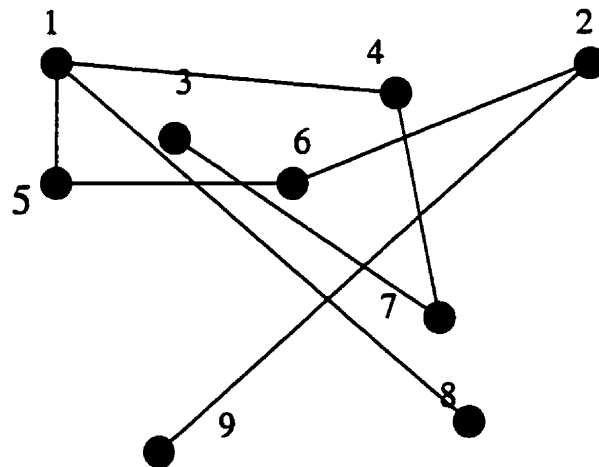
Figure 5.6: *Topologie du réseau en arbre optimal.*

Tableau 5.7: Variables d'arcs.

s	(i, j)	N_s	\tilde{N}_s	a_s	\hat{a}_s	B_s	\tilde{B}_s
3	(1,4)	18.5	18.8	8.92	9.00	0.0018	0.0017
4	(1,5)	21.6	21.6	10.90	11.00	0.0014	0.0017
7	(1,8)	14.0	14.1	5.95	6.00	0.0022	0.0020
12	(2,6)	15.5	15.8	6.94	7.00	0.0020	0.0017
15	(2,9)	15.5	15.8	6.94	7.00	0.0020	0.0017
19	(3,7)	10.4	11.0	3.97	4.00	0.0033	0.0020
24	(4,7)	18.5	18.8	8.92	9.00	0.0018	0.0017
27	(5,6)	15.5	15.8	6.94	7.00	0.0020	0.0017

concentration de trafic. Nous obtenons le RAQO de la figure 5.7. Le coût du réseau est 24.5 et le revenu est 132.8. Ainsi, nous avons encore réduit le coût du réseau de 17.5% par rapport à celui de l'exemple 5.2.2. Ces résultats sont résumés dans le tableau 5.8.

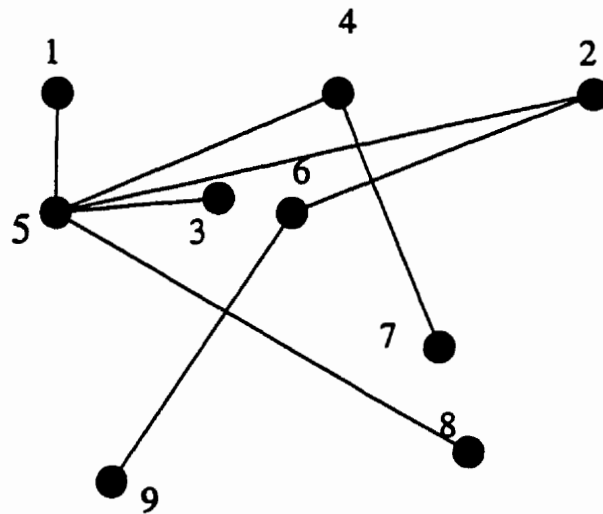
La QS est donné dans le tableau 5.9. La fonction objectif vaut $z = -108.3$ et nous avons $\tilde{z} = -107.9$. Les variables d'arc sont données dans le tableau 5.10. Comme nous pouvons le constater d'après ces résultats, nos approximations sont très bonnes dans ce cas aussi.

5.4.5 Concentration individuelle du trafic

On peut montrer [71, 70] que la procédure décrite dans la section 5.4.4 est quasi-optimale pour des réseaux avec des communication point-à-multipoint. Cependant, il se peut que pour quelques types, on puisse générer un arbre différent de celui obtenu

Tableau 5.8: *Synthèse de réseaux avec du multi-débit.*

	Coût	Revenu	Valeur nette	$\max_k L^k$
Réf. (étoile,MST,Steiner)	29.7	132.7	103.0	.011
Concentration de trafic	24.5	132.8	108.3	.010

Figure 5.7: *Topologie du RAQO avec du multi-débit.*Tableau 5.9: *QS pour la synthèse du réseau avec du multi-débit.*

(k, c)	(1, 1)	(2, 1)	(3, 1)	(1, 2)	(2, 2)
$L^k(c)$	0.007	0.008	0.006	0.011	0.010
$\tilde{L}^k(c)$	0.009	0.010	0.009	0.010	0.009

Tableau 5.10: Variables d'arcs pour le multi-débit (classe 2 seulement).

s	(i, j)	N_s	\tilde{N}_s	$a_s(2)$	$\hat{a}_s(2)$	$B_s(2)$	$\tilde{B}_s(2)$
4	(1,5)	20.8	21.5	0.00	0.00	0.0000	0.0000
11	(2,5)	39.8	39.6	6.94	7.00	0.0017	0.0020
12	(2,6)	30.4	31.0	3.97	4.00	0.0024	0.0020
17	(3,5)	35.9	36.1	6.94	7.00	0.0019	0.0020
22	(4,5)	29.3	30.0	2.98	3.00	0.0032	0.0025
24	(4,7)	29.3	30.0	2.98	3.00	0.0031	0.0025
29	(5,8)	29.1	29.7	3.97	4.00	0.0025	0.0020
33	(6,9)	30.4	31.0	3.97	4.00	0.0024	0.0020

à l'aide de la procédure décrite ci-dessus et obtenir encore de meilleurs résultats. On parle alors d'une concentration *individuelle* de trafic, en considérant les types d'appel un par un et ce, après avoir effectué la concentration globale de trafic. Avec la concentration individuelle du trafic, même si on a des appels point-à-point, le réseau obtenu est quasi-optimal, ce qui n'est pas vrai si on se contente de la procédure décrite dans la section 5.4.4.

La concentration individuelle de trafic est intéressante lorsque pour certains types on a un petit nombre de nœuds participants, typiquement, des connexions point-à-point. Souvent, pour ce type d'appel, l'arbre obtenu par la procédure décrite dans la section 5.4.4 est un arbre de Steiner de grande taille (avec plusieurs nœuds de Steiner). Remplacer cet arbre par un arbre plus petit (un arbre de recouvrement) donne souvent de meilleurs résultats. Ainsi, nous concentrons davantage le trafic en réduisant le nombre d'arcs. Néanmoins, dans la mesure où la concentration individuelle

du trafic concerne un petit nombre de types d'appels, les économies réalisées sont négligeables par rapport aux économies déjà effectuées par la concentration globale de trafic. Ainsi, nous pouvons voir la concentration globale comme étant *grossière* alors que la concentration individuelle peut être vue comme étant *fine*.

Un exemple est donné dans la figure 5.8 où nous avons deux cas. Un cas où le trafic de A vers B est envoyé sur le chemin utilisant les trois arcs s_1 , s_2 , et s_3 (nous supposons que ces arcs appartiennent au RAQO). Dans l'autre cas, nous utilisons l'arc direct s de A vers B (cet arc n'est pas dans le RAQO). Le coût du réseau est réduit avec une telle opération si

$$N_{s_1} + N_{s_2} + N_{s_3} > N'_s + N'_{s_1} + N'_{s_2} + N'_{s_3} \quad (5.14)$$

où les variables primées correspondent à l'utilisation de l'arc direct au lieu du chemin s_1 , s_2 , et s_3 .

L'inégalité (5.14) est susceptible d'être satisfaite si la demande de trafic de A vers B est assez forte pour que nous ayons $N'_s \ll N_s, \forall i \in \{1, 2, 3\}$. Cependant, si cette demande de trafic est très forte, l'arc s a de fortes chances de faire partie du RAQO ce qui veut dire que nous n'avons pas besoin de la concentration individuelle de trafic dans ce cas.

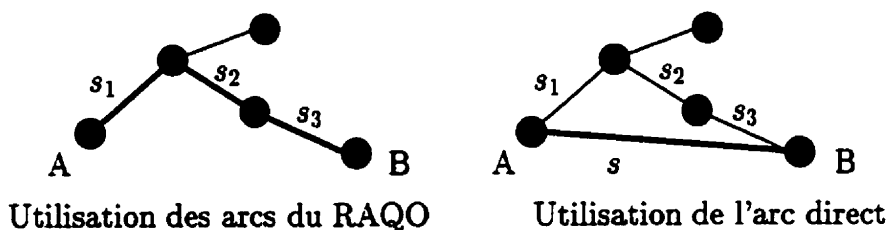


Figure 5.8: *Utilisation de l'arc direct au lieu des arcs du RAQO.*

Ainsi, après la génération du RAQO, on effectue la procédure suivante

1. $k = 0$, $m = 1$. Pour tous les arcs du réseau non utilisés, $\ell_s = -1$.
2. $k = k + 1$, si $k = K + 1$, fin.
3. Si l'arbre (k, c, m) , généré sur l'arbre optimal n'est pas un arbre de recouvrement, générer un arbre de recouvrement de coût maximum à sa place, en utilisant ℓ_s comme longueur d'arcs. Sinon, aller à (2).
4. Si on améliore la fonction objectif (valeur nette évaluée approximativement tel que indiqué dans la section 5.4.1), remplacer l'arbre de Steiner par l'arbre de recouvrement dans $S^k(c)$.
5. Aller à (2).

Au début de cette procédure, les ℓ_s des arcs n'appartenant pas au RAQO doivent prendre des valeurs négatives afin de donner la préférence aux arcs qui sont déjà sélectionnés dans le RAQO. Une fois que nous avons sélectionné les arbres, nous optimisons les capacités à l'aide de l'algorithme de la section 3.4.2.

À la fin de cette procédure, on a toujours un arbre unique pour connecter un type d'appel donné mais la topologie optimale du réseau n'est pas nécessairement un arbre. Ceci n'est pas forcément mauvais compte tenu de l'introduction d'une certaine sécurité dans le réseau d'une part, et de la possibilité d'optimiser l'acheminement si le trafic change (voir section 5.6), d'autre part.

Exemple 5.4.3 Dans cet exemple, aux données de l'exemple 5.4.1, nous ajoutons un type d'appel (type 4) point-à-point du nœud 6 vers le nœud 8. On suppose que $A^4 = 4$. D'abord, nous n'utilisons pas la concentration individuelle du trafic et nous

cherchons le RAQO. Nous obtenons un réseau dont la topologie est exactement celle donnée dans la figure 5.6 mais dont les capacités de certains arcs ont augmenté. Celles-ci sont données dans le tableau 5.11 où on les compare avec les capacités de référence obtenue dans l'exemple 5.4.1. Le coût du réseau est passé de 13 pour le réseau de référence à 14.6 ($C_0 = 0.1$).

Nous pouvons constater que les capacités des arcs (1, 5), (1, 8) et (5, 6) ont augmenté. Ces arcs correspondent au chemin utilisé pour connecter les appels de 6 vers 8 dans le RAQO de la figure 5.6. Ainsi, nous avons deux nœuds de transit dans ce chemin ce qui résulte en un gaspillage de capacité. En effet, si nous utilisons la concentration individuelle du trafic, nous obtiendrions un réseau identique à celui la figure 5.6 mais auquel nous ajouterions l'arc (6, 8). Les capacités des arcs sont exactement celles données dans le tableau 5.7 et la capacité de l'arc (6, 8) est $N_{32} = 9.3$ (nous avons aussi $\tilde{N}_{32} = 9.3$). Le coût du nouveau réseau est 13.9 (au lieu de 14.6 sans la concentration individuelle) ce qui veut dire que nous avons fait environ 5% d'économies.

5.4.6 Dilemme point-à-point/multipoint

Nous avons constaté que l'introduction du point-à-multipoint avec le point-à-point donne des réseau dont la connectivité varie selon l'intensité du trafic de l'un ou de l'autre [71]. Plus le trafic point-à-point est intense par rapport au trafic point-à-multipoint, plus l'utilisation des arcs directs est optimale et le trafic point-à-multipoint est alors acheminé sur les arcs déjà utilisés pour le trafic point-à-point. Ainsi, si nous avons plusieurs appels point-à-point, la topologie est relativement fortement connectée. Dans le cas contraire, où le trafic point-à-multipoint est plus

Tableau 5.11: *Capacités des arcs avec un appel point-à-point et trois appels point-à-multipoint.*

s	(i, j)	N_s de référence	N_s
3	(1,4)	18.5	18.6
4	(1,5)	21.6	26.9
7	(1,8)	14.0	19.6
12	(2,6)	15.5	15.6
15	(2,9)	15.5	15.6
19	(3,7)	10.4	10.5
24	(4,7)	18.5	18.6
27	(5,6)	15.5	21.1

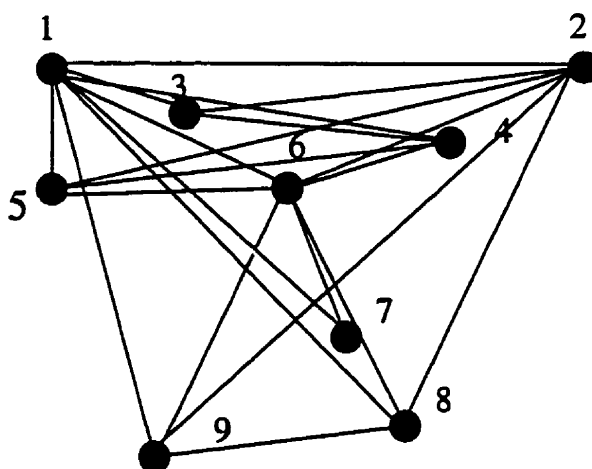
intense, on a tendance à obtenir des topologies faiblement connectées et le trafic point-à-point est alors acheminé sur des chemins avec un nombre plus ou moins élevé de nœuds de transit. Dans le cas général, la connectivité du réseau optimal varie entre celle d'un arbre et celle d'un graphe complet.

Exemple 5.4.4 L'objectif de cet exemple est d'illustrer d'une part, l'utilité de la concentration individuelle du trafic et d'autre part, l'impact de l'introduction d'appels point-à-point sur la topologie optimale du réseau. Pour ce faire, aux données de l'exemple 5.2.1, nous ajoutons sept types d'appel point-à-point. Ceux-ci sont donnés dans le tableau 5.12.

Considérons d'abord le cas où nous avons une sélection d'arbres dans la quelle on utilise un arbre unique en étoile pour connecter tous les appels d'un type donné. Un arbre en étoile peut être considéré comme un arbre « direct » dans la mesure où

Tableau 5.12: *Types d'appel point-à-point.*

k	source	destinations	w^k	A^k
4	2	5	1.00	4.00
5	7	1	1.00	6.00
6	5	4	1.00	5.00
7	8	9	1.00	4.00
8	9	1	1.00	5.00
9	2	3	1.00	4.00
10	6	8	1.00	5.00

Figure 5.9: *Topologie optimale du réseau avec des appels point-à-point en utilisant les arbres directs.*

on utilise l'arc direct entre la source et chaque destination. Nous dimensionnons le réseau à l'aide de l'algorithme de la section 3.4.2 et nous obtenons le réseau de la figure 5.9. Le coût de ce réseau est de 22.8 ($C_0 = 0.1$) et le revenu des appels est 102.0 et $L^k \leq 0.011 \forall k$.

Maintenant, nous optimisons la sélection d'arbres à l'aide de l'algorithme que nous venons de présenter ci-dessus c.-à-d., on fait d'abord une concentration globale du trafic puis une concentration individuelle. Ceci nous donne le réseau optimal de 11 arcs de la figure 5.10 dont le coût est 18.8. Le revenu généré par les appels est 102.2 et $L^k \leq 0.011 \forall k$. Ainsi, nous avons réalisé des économies de 18% par rapport au cas où nous utilisons les étoiles.

D'autre part, on voit ici que l'introduction d'appels point-à-point fait en sorte que la topologie optimale (voir figure 5.10) obtenue par concentration du trafic n'est plus un arbre. Si nous n'avions pas utilisé la concentration individuelle de trafic, la topologie optimale du réseau aurait été un arbre de coût 22.0, ce qui signifie qu'avec la concentration individuelle de trafic, nous avons réduit le coût du réseau de 15% par rapport au cas où l'on effectue seulement la concentration globale du trafic. Les résultats sont résumés dans le tableau 5.13.

Il est aussi intéressant de noter que tous les types point-à-point utilisent les arcs directs et que le trafic multipoint est écoulé sur ces mêmes arcs. Nous avons constaté que si nous diminuons l'intensité du trafic point-à-point, le réseau optimal est un arbre (la concentration individuelle du trafic n'est pas requise) et le trafic point-à-point n'est pas nécessairement écoulé sur les arcs directs ce qui illustre le dilemme point-à-point/multipoint (voir exemple 5.4.8).

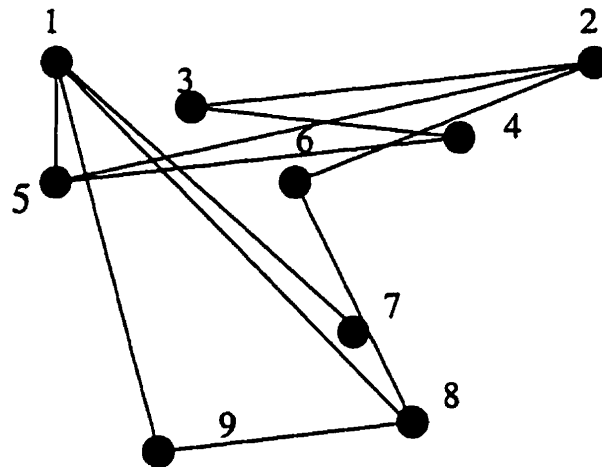


Figure 5.10: *Topologie optimale du réseau avec des appels point-à-point.*

Tableau 5.13: *Synthèse de réseaux avec point-à-point et point-à-multipoint.*

	Coût	Revenu	Valeur nette	$\max_k L^k$
Référence (étoiles)	22.8	102.0	79.2	.011
Concentration globale	22.0	102.2	80.2	.011
Concentration individuelle	18.8	102.1	83.3	.011

5.4.7 Résultats pour d'autres réseaux

Étant donné que la concentration globale de trafic est intéressante si on a plusieurs appels point-à-multipoint et que la concentration individuelle l'est plutôt quand on a des appels point-à-point, nous voulons alors savoir dans quelle mesure on économise si on a les deux catégories d'appel dans le réseau.

Nous allons considérer comme référence un réseau obtenu après dimensionnement en utilisant des étoiles pour connecter les appels point-à-multipoint (une étoile par type d'appel). Nous définissons alors η_g comme étant l'économie (en pourcent) réalisée en utilisant l'algorithme de concentration globale de trafic et η_i comme étant l'économie additionnelle réalisée à l'aide de la concentration individuelle. L'économie totale est alors donnée par $\eta = \eta_g + \eta_i - \eta_g\eta_i/100$.

Nous voulons savoir comment η_g , η_i et η évoluent en fonction du nombre de nœuds N dans le réseau, du nombre de types d'appel K , du rapport ρ défini comme étant la fraction du nombre de type d'appel point-à-point sur le nombre de types d'appel point-à-multipoint et du nombre de classes C .

Exemple 5.4.5 Dans cet exemple, nous considérons une seule classe d'appel et nous faisons varier N , K et ρ . Les Résultats sont donnés dans la figure 5.11 pour η_g , dans la figure 5.12 pour η_i et dans la figure 5.13 pour η .

D'après les résultats de la figure 5.11, nous constatons que η_g augmente avec la taille du réseau. Ceci est dû au fait que lorsqu'on a plusieurs nœuds, utiliser les arbres directs (étoiles) donne lieu à un réseau très connecté qui coûte très cher. En effet, comme on peut le voir dans la figure 5.11, pour $N = 25$, $K = 20$ et $\rho = 0/1$, nous avons une économie de 45% en concentrant le trafic sur un arbre (RAQO). D'autre part, nous constatons que dans la mesure où la proportion d'appels point-à-point

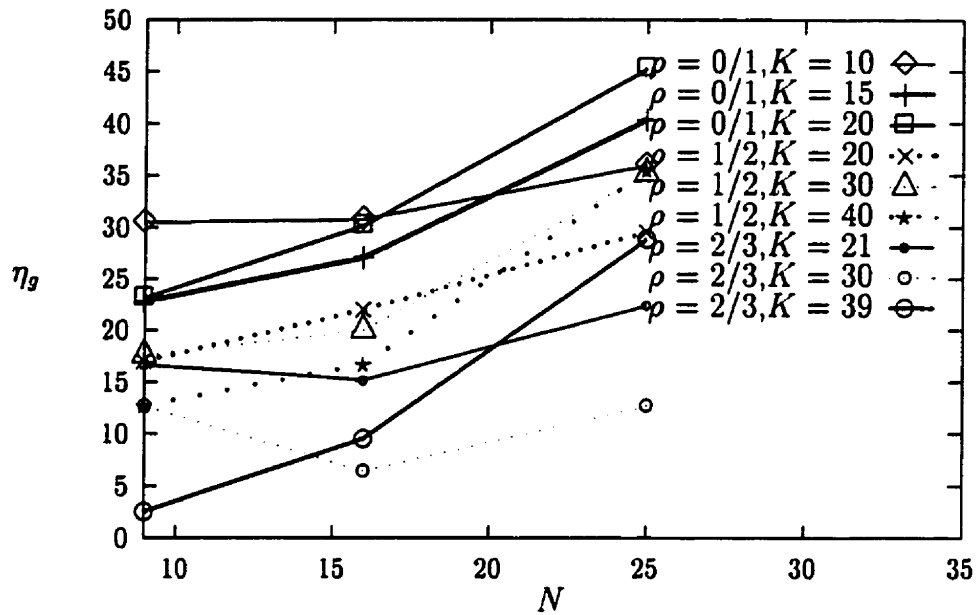


Figure 5.11: Variation de η_g (en %) vs N .

diminue (indiquée par ρ), les économies réalisées par concentration globale du trafic augmentent. Ceci est dû au fait que si on a plusieurs appels point-à-multipoint, on gagne beaucoup à concentrer le trafic.

D'autre part, d'après les résultats de la figure 5.12, nous constatons que η_i ne dépend pas particulièrement du nombre de nœuds ou de types d'appels. En revanche, η_i augmente considérablement avec ρ et ceci est dû à l'économie réalisée par la concentration individuelle pour le trafic point-à-point.

Enfin, d'après la figure 5.13, on peut voir que η donne des économies importantes dans pratiquement tous les cas excepté lorsqu'on a plusieurs types d'appel point-à-point (ρ et K grands). Ceci est dû au fait que dans ce cas, nous n'avons pas beaucoup d'économies avec la concentration globale de trafic et des économies relativement faibles par la concentration individuelle du trafic. En effet, si nous utilisons les arbres

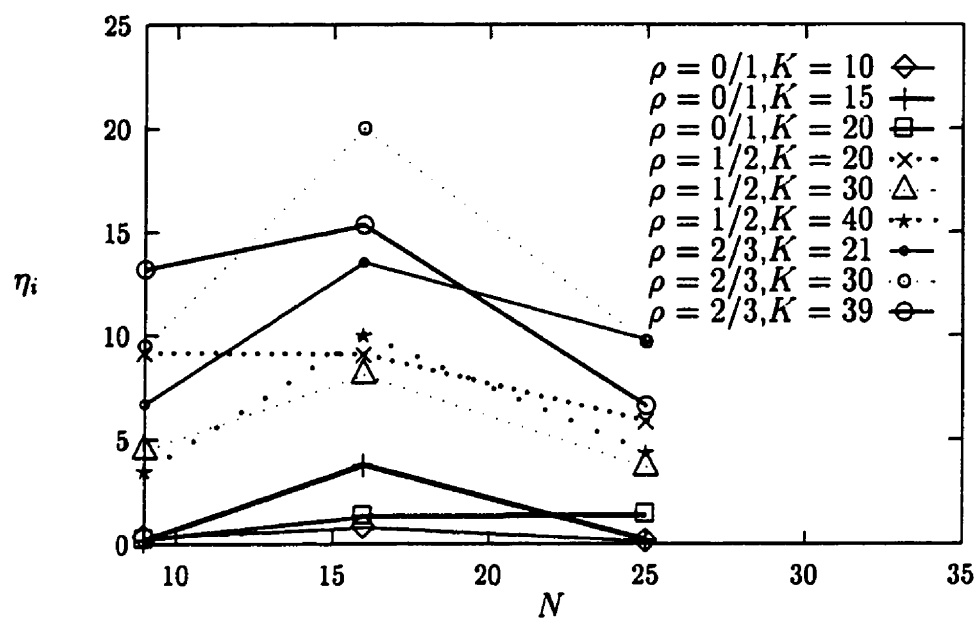


Figure 5.12: Variation de η_i (en %) vs N .

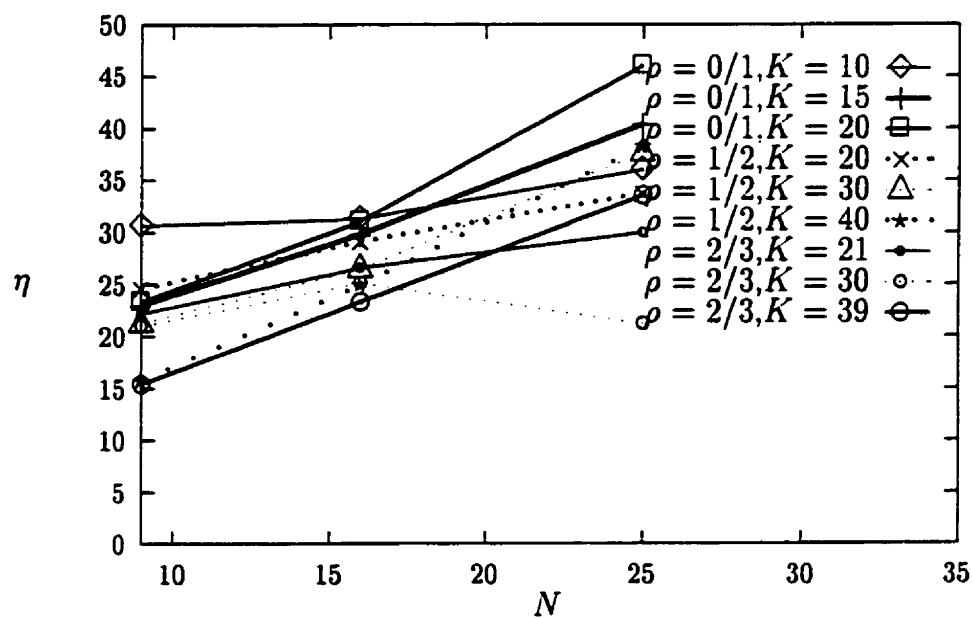


Figure 5.13: Variation de η (en %) vs N .

directs comme référence, les arcs directs correspondant à la plupart des appels point-à-point sont déjà utilisés. Ceci est illustré dans la figure 5.13 pour $N = 9$, $K = 40$ et $\rho = 1/2$ et pour $N = 9$, $K = 39$ et $\rho = 2/3$ où η est de l'ordre de 15%.

Exemple 5.4.6 Dans cet exemple, nous étudions l'évolution de η_g , η_i et η en fonction du nombre de classes d'appel. Pour cette fin, nous considérons un réseau de 9 nœuds avec 15 types d'appel en tout. Nous considérons trois cas avec respectivement, une deux et trois classes ayant chacune $W(1) = 1$, $W(2) = 2$ et $W(3) = 4$ comme débits. Dans chaque cas, nous avons deux réseaux avec $\rho = 1/3$ et $\rho = 2/3$.

Nous supposons que le trafic point-à-point utilise une faible largeur de bande car en pratique, ce trafic correspond à de la voix et à des données informatiques alors que le multipoint correspond à du trafic vidéo et multimédia.

Dans le tableau 5.14, nous donnons (η_g, η_i, η) et dans le tableau 5.15, nous donnons la QS. Ainsi, on peut voir que la concentration du trafic permet d'économiser davantage dans le cas du multi-débit et ceci est dû au volume important du trafic dans ce cas. En effet, les économies augmentent avec le nombre de classes surtout lorsqu'on a plusieurs appels point-à-multipoint comme c'est le cas pour $\rho = 1/3$ dans le tableau 5.14.

D'autre part, nous pouvons voir d'après le tableau 5.14 que lorsque nous avons plus d'une classe d'appel, la concentration individuelle du trafic n'est pas requise ($\eta_i = 0$). Ceci veut dire que tout le trafic point-à-point est acheminé sur les arcs utilisés pour le multipoint. Ainsi, dans le cas de plusieurs classes, si l'arc direct correspondant à un appel point-à-point donné n'est pas dans le RAQO, ce dernier est acheminé sur un chemin à plusieurs nœuds de transit. Ceci est dû au fait que dans ce cas, les grosses capacités des arcs font en sorte qu'il est plus économique de les

Tableau 5.14: (η_g, η_i, η) pour des réseaux multi-débit.

ρ	Une classe	Deux classes	Trois classes
1/3	(8.6, 1.5, 10.0)	(28.9, 0, 28.9)	(30.3, 0, 30.3)
2/3	(14.9, 9.9, 23.3)	(22.9, 0, 22.9)	(23.9, 0, 23.9)

augmenter légèrement pour acheminer le trafic point-à-point plutôt que de mettre la capacité sur l'arc direct, comme on pouvait s'y attendre d'après la proposition 5.4.1. En effet, comme on peut le voir dans le tableau 5.15, les probabilités de blocage des appels de faible débit sont très petites par rapport à celles des appels à haut débit et ceci explique pourquoi on a intérêt à utiliser les arcs du RAQO seulement. Ce phénomène est dû au fait que nous n'avons pas de réservation de circuit dans notre modèle de synthèse. Si tel est le cas, les résultats peuvent être différents [96].

Enfin, les capacités optimales et les variables d'arcs sont données dans le tableau 5.16 pour $\rho = 2/3$ et la classe 3 dans le cas où nous avons trois classes d'appel. On peut alors vérifier que nos approximations sont bonnes dans cet exemple aussi.

Tableau 5.15: $\max_k L^k(c)$ obtenues par concentration du trafic pour des réseaux multi-débit.

	Une classe		Deux classes		Trois classes	
ρ	1/3	2/3	1/3	2/3	1/3	2/3
$c = 1$	0.009	0.010	0.005	0.003	0.001	0.001
$c = 2$	—	—	0.011	0.011	0.005	0.003
$c = 3$	—	—	—	—	0.011	0.011

Tableau 5.16: Variables d'arcs pour la classe 3 avec trois classes d'appel et $\rho = 2/3$.

s	(i, j)	N_s	\tilde{N}_s	$a_s(3)$	$\hat{a}_s(3)$	$B_s(3)$	$\tilde{B}_s(3)$
1	(1,2)	102.9	105.3	10.92	11.00	0.0021	0.0014
7	(1,8)	110.6	111.7	9.92	10.00	0.0016	0.0014
8	(1,9)	115.4	116.1	12.89	13.00	0.0013	0.0014
13	(2,7)	73.9	74.4	5.95	6.00	0.0016	0.0014
20	(3,8)	83.8	85.4	6.95	7.00	0.0020	0.0017
23	(4,6)	72.0	72.2	7.93	8.00	0.0014	0.0014
26	(4,9)	109.6	110.2	10.91	11.00	0.0014	0.0014
30	(5,9)	90.2	92.0	9.92	10.00	0.0017	0.0014

Exemple 5.4.7 Dans cet exemple, nous allons considérer des données particulières qui vont nous permettre d'observer l'évolution des économies obtenues par concentration de trafic en fonction du nombre de nœuds N et de l'intensité du trafic offert (les A^k) et ce, pour des grands réseaux. Cependant, il est très difficile d'obtenir des résultats pour un très grand nombre de types d'appel ($K \geq 50$) et un très grand réseau ($N \geq 50$) avec les outils informatiques dont nous disposons (pas assez de mémoire). Néanmoins, les approximations que nous donnons dans la section 5.4.1 peuvent être utilisées pour n'importe quelles valeurs de N et de K .

L'objectif ici est de donner une idée sur l'évolution des économies pour N grand, alors, les données que nous allons utiliser ici seront choisies de sorte qu'elles permettent d'isoler l'effet de N et des A^k (il va sans dire que si N est grand, en pratique, K est grand aussi).

Nous supposons que nous avons un réseau de N nœuds et que nous avons $K = N$

types d'appel point-à-multipoint. Ainsi, nous allons nous intéresser seulement à η_g . Pour simplifier, nous considérons une seule classe d'appel et nous supposons que chaque nœud du réseau joue le rôle d'une source dans un type d'appel différent et nous supposons alors qu'un nœud i donné est la source du type d'appel $k = i$. Nous supposons que chaque type d'appel $k = i$ possède $N - 1$ destinations (tous les nœuds $j \neq i$). De plus, nous supposons que nous avons la même demande de trafic $A^k = A_m \forall k$ pour tous les types d'appel.

Nous allons d'abord supposer que nous utilisons un arbre en étoile pour connecter chaque type d'appel. Il est alors clair que si nous dimensionnons le réseau, nous obtenons un graphe complet ayant $N(N - 1)/2$ arcs. Chaque arc s connectant les nœuds (i, j) aura alors comme trafic offert la somme des trafics offerts provenant du type d'appel i (de i vers j) et du type d'appel j (de j vers i). Comme chaque type a une demande de trafic A_m , on a $\hat{a}_s = 2A_m \forall s$. En supposant que nous dimensionnons le réseau pour une QS donnée par \bar{L}^k , comme nous avons $(N - 1)$ destinations, nous pouvons calculer les \bar{B}_s d'après l'approximation (5.6) et les \bar{N}_s d'après l'approximation (5.7).

Maintenant, nous allons supposer que nous utilisons un RAQO. Il faut noter que dans la mesure où les types d'appels ont les mêmes demandes de trafic et que chacun d'entre eux implique tous les nœuds du réseau, tous les arbres qui recouvrent les nœuds du réseau ont des performances identiques en terme de coût du réseau. En effet, le trafic potentiel qui pourrait être offert à tous les arcs est le même et vaut $\ell_s = NA_m \forall s$. Il s'agit en fait de la somme des trafics offerts provenant des N types d'appel. Alors, de même que ci-dessus, en supposant que nous dimensionnons le réseau pour une QS donnée par \bar{L} , comme nous avons $(N - 1)$ destinations, nous

pouvons calculer les \tilde{B}_s d'après l'approximation (5.6) et celles-ci auront les mêmes valeurs que lorsqu'on utilise les étoiles. Alors, on peut déterminer les \tilde{N}_s en utilisant l'approximation (5.7).

Nous pouvons alors approximer l'économie réalisée en utilisant le RAQO au lieu des étoile en observant que dans le premier cas, nous utilisons $N(N-1)/2$ arcs de capacité $f(2A_m)$ chacun et que dans le deuxième cas, nous utilisons seulement $N-1$ arcs de capacité $f(NA_m)$ cette fois; où f est la fonction donnée par l'équation (5.10) avec $B_s = Ct. = \bar{L}/D$.

Dans la mesure où f est concave, on a $Nf(2A_m) > 2f(NA_m)$ (voir lemme B.1.2 donné dans l'annexe B) et on a ainsi

$$C_0 \frac{N(N-1)}{2} f(2A_m) > C_0 (N-1) f(NA_m). \quad (5.15)$$

Le terme de droite de l'équation (5.15) donne le coût approximatif dans le cas où nous utilisons les étoiles et le terme de gauche, dans le cas où nous utilisons le RAQO. Ainsi, les économies réalisées sont

$$\eta'_g = \frac{Nf(2A_m) - 2f(NA_m)}{Nf(2A_m)}. \quad (5.16)$$

Afin de vérifier le bien fondé de l'équation (5.16), nous avons comparé ses résultats à ceux obtenus pour η_g dans la figure 5.11 pour respectivement $N = 9, 16$ et 25. Nous obtenons alors les résultats du tableau 5.17. On peut remarquer que même si les résultats obtenus dans la figure 5.11 ne correspondent pas à des données similaires à celles que nous utilisons dans cet exemple, les résultats de la figure 5.11 et ceux obtenus par l'équation (5.16) sont du même ordre de grandeur. Cela nous indique que l'équation (5.16) est susceptible de donner une bonne estimation des économies obtenues par concentration de trafic. Notons que pour évaluer η'_g , nous

Tableau 5.17: Comparaison de η_g et η'_g .

N	η_g (%)	η'_g (%)
9	30 ($K = 10$)	33 ($K = 9$)
16	27 ($K = 15$)	39 ($K = 16$)
25	45 ($K = 20$)	43 ($K = 25$)

avons supposé que $A_m = 4$ ce qui correspond, à peu de choses près, à la demande de trafic moyenne pour les données de la figure 5.11 et que $B_s = Ct. = 0.001$.

Maintenant, nous voulons observer l'évolution de η'_g en fonction de N . Ceci est donné dans la figure 5.14 pour différentes valeurs de A_m . Nous avons supposé que $\bar{L} = 0.01$ et donc que $B_s = Ct. = 0.01/(N - 1)$.

Comme on peut le constater d'après cette figure, l'économie augmente rapidement pour N relativement moyen (entre 0 et 25) pour ne plus croître que de façon modérée à partir de $N = 40$. Néanmoins, l'économie est autour de 50% pour de grandes valeurs de N ce qui veut dire que nous avons encore plus d'intérêt à concentrer le trafic pour des grands réseaux. Ainsi, nous pouvons dire que dès que N dépasse 25, on peut s'attendre à ce qu'en pratique, les économies obtenues par concentration de trafic soient très importantes.

D'autre part, nous constatons que les économies sont plus importantes si les demandes de trafic (A_m) sont faibles. Ceci est dû au fait qu'alors, nous sommes dans la région de courbure forte de la fonction f (voir figure 5.4) et les économies obtenues par concentration du trafic sont très importantes.

Exemple 5.4.8 De même que dans l'exemple 5.4.7, nous allons ici considérer des

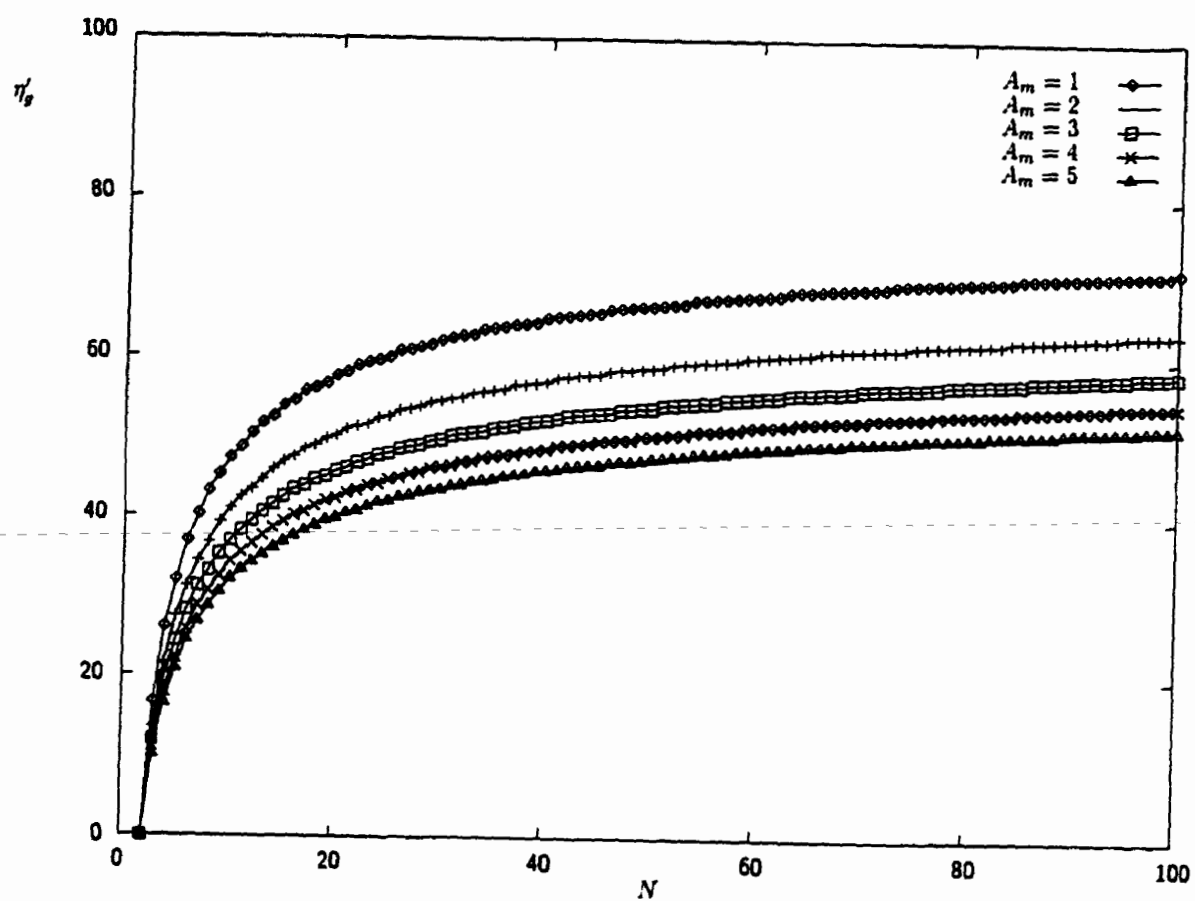


Figure 5.14: η'_g en fonction de N .

données particulières qui vont nous permettre d'observer l'évolution des économies en fonction de N et des A^k pour des grands réseaux mais cette fois, en présence de trafic point-à-point aussi. On s'intéresse donc à η_i et η_g .

Aux données de l'exemple 5.4.7, nous ajoutons des appels point-à-point entre chaque paire de nœuds. Nous supposons que les appels point-à-point ont tous les mêmes demandes de trafic A_p et que les appels point-à-multipoint ont tous les mêmes demandes de trafic A_m . Parce que nous voulons entre autre étudier l'impact du rapport A_p/A_m sur η_i et η_g , nous allons supposer que $A_m = 1$ et nous allons évaluer η_i et η_g en fonction de A_p .

Nous considérons d'abord le cas où nous utilisons les arbres et les arcs directs pour connecter tous les types d'appel. Cela résulte en un graphe complet ($N(N-1)/2$ arcs) où le trafic offert à chaque arc est $\hat{a}_s = 2A_m + A_p \forall s$. En effet, puisque chaque appel point-à-point utilise son arc direct, nous ajoutons A_p au trafic point-à-multipoint que nous avions dans l'exemple 5.4.7.

Maintenant, nous utilisons un RAQO. Nous avons $\ell_s = NA_m + A_p \forall s$. Ainsi, on a plusieurs RAQO. Pour simplifier, nous allons supposer que nous utilisons un RAQO en étoile. Nous allons noter i_0 la racine du RAQO. Cela veut dire que $(N-1)$ appels point-à-point utiliseront les arcs directs (ceux ayant i_0 comme source ou destination) et $N(N-1)/2 - (N-1)$ appels point-à-point utiliseront des chemins à un nœud de transit (le nœud de transit est i_0). Ainsi, on peut montrer que le trafic offert à chaque arc (i_0, i) du réseau est $\hat{a}_s = NA_m + (N-1)A_p \forall s$. Ceci est le trafic multipoint de l'exemple 5.4.7, auquel on ajoute le trafic correspondant à $N-1$ appels point-à-point impliquant le nœud i .

Maintenant, nous supposons que nous utilisons la concentration individuelle du

trafic. Nous supposons que seuls les appels point-à-point subiront une concentration individuelle. Alors, nous auront $N-1$ arcs auxquels le trafic offert est $\hat{a}_s = NA_m + A_p$ (il s'agit du trafic multipoint auquel est ajouté le trafic point-à-point offert à l'arc direct); et $N(N-1)/2 - (N-1)$ arcs auxquels le trafic offert est $\hat{a}_s = A_p$ (seulement du trafic point-à-point).

Il faut noter ici que nous ne savons pas à priori si nous aurons des économies par concentration du trafic car cela dépend de la valeur de A_p comme on va le voir tout de suite sur des exemples numériques. Nous pouvons alors montrer que l'économie par concentration globale du trafic est

$$\eta'_g = \max \left[0, \frac{Nf(2A_m + A_p) - 2f(NA_m + (N-1)A_p)}{Nf(2A_m + A_p)} \right] \quad (5.17)$$

et que l'économie par concentration individuelle du trafic est

$$\eta'_i = \max \left[0, \frac{f(NA_m + (N-1)A_p) - (f(NA_m + A_p) + (N/2 - 1)f(A_p))}{f(NA_m + (N-1)A_p)} \right]. \quad (5.18)$$

Les résultats sont donnés dans la figure 5.15 pour η'_g et dans la figure 5.16 pour η'_i . Nous pouvons alors constater d'après la figure 5.15 que η'_g augmente avec N et qu'elle diminue avec A_p . Cela est prévisible dans la mesure où plus le trafic point-à-point est intense, moins il est avantageux d'effectuer la concentration globale. Néanmoins, pour de grands réseaux ($N > 50$), l'économie est au delà de 20% même si le trafic point-à-point est cinq fois plus intense que le trafic point-à-multipoint (voir courbe $A_p = 5$ dans la figure 5.15). Cela est dû au nombre très grand de destinations dans les appels point-à-multipoint.

D'autre part, d'après la figure 5.16, on constate que η'_i semble croître très rapidement pour des petites valeurs de N pour atteindre un maximum pour un certain N (autour de 5) et commence à décroître pour devenir négligeable quand N est très

grand. Notons que nous avons un comportement similaire pour les résultats obtenus dans la figure 5.12. On peut alors interpréter ce résultat en considérant deux aspects 1) quand N est très grand, le trafic point-à-multipoint est très fort (NA_m) ce qui minimise l'importance de la concentration individuelle du trafic alors que 2) si N est assez faible (autour de 5), nous avons des valeurs comparables entre A_p et NA_m et alors, on a des économies relativement importante par concentration individuelle du trafic. D'autre part, nous constatons que la concentration individuelle du trafic n'est pertinente que si le trafic point-à-point est au moins 3 fois plus fort que le trafic point-à-multipoint. Cela veut dire que si A_p est faible, utiliser un RAQO est optimal et il est plus avantageux de connecter les appels point-à-point sur des chemins avec un nombre de nœuds de transit plus ou moins grand.

5.5 Variation du trafic

Nous voulons étudier le problème de sensibilité de la performance du réseau en présence de variation de trafic due à des erreurs de prévision soit sur les demandes de trafic ou sur les ensembles de nœuds participants dans les communications point-à-multipoint. On sait que dans le cas où on a uniquement des connexions point-à-point, le réseau est relativement peu sensible aux variations de trafic grâce en particulier aux techniques d'acheminement adaptatif. Cependant, pour des connexions point-à-multipoint, la question n'est pas résolue.

Nous voulons alors savoir comment un opérateur du réseau doit réagir vis-à-vis de la variation de trafic. Nous avons considéré quatre façons de réagir:

1. Une façon triviale de réagir à la variation de trafic est de garder le même réseau, avec les mêmes capacités et la même sélection d'arbres (si possible) et ne rien

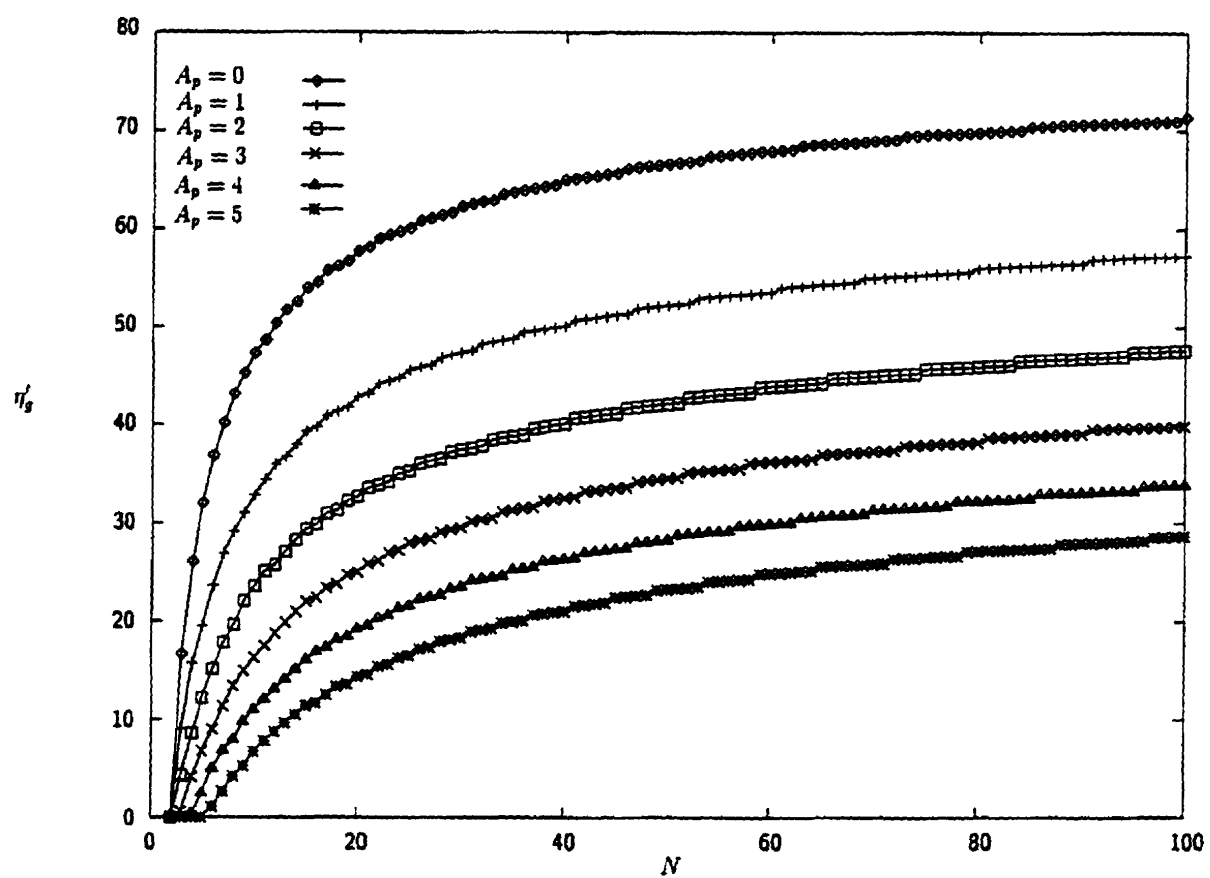


Figure 5.15: η'_g (en %) en fonction de N avec du point-à-point.

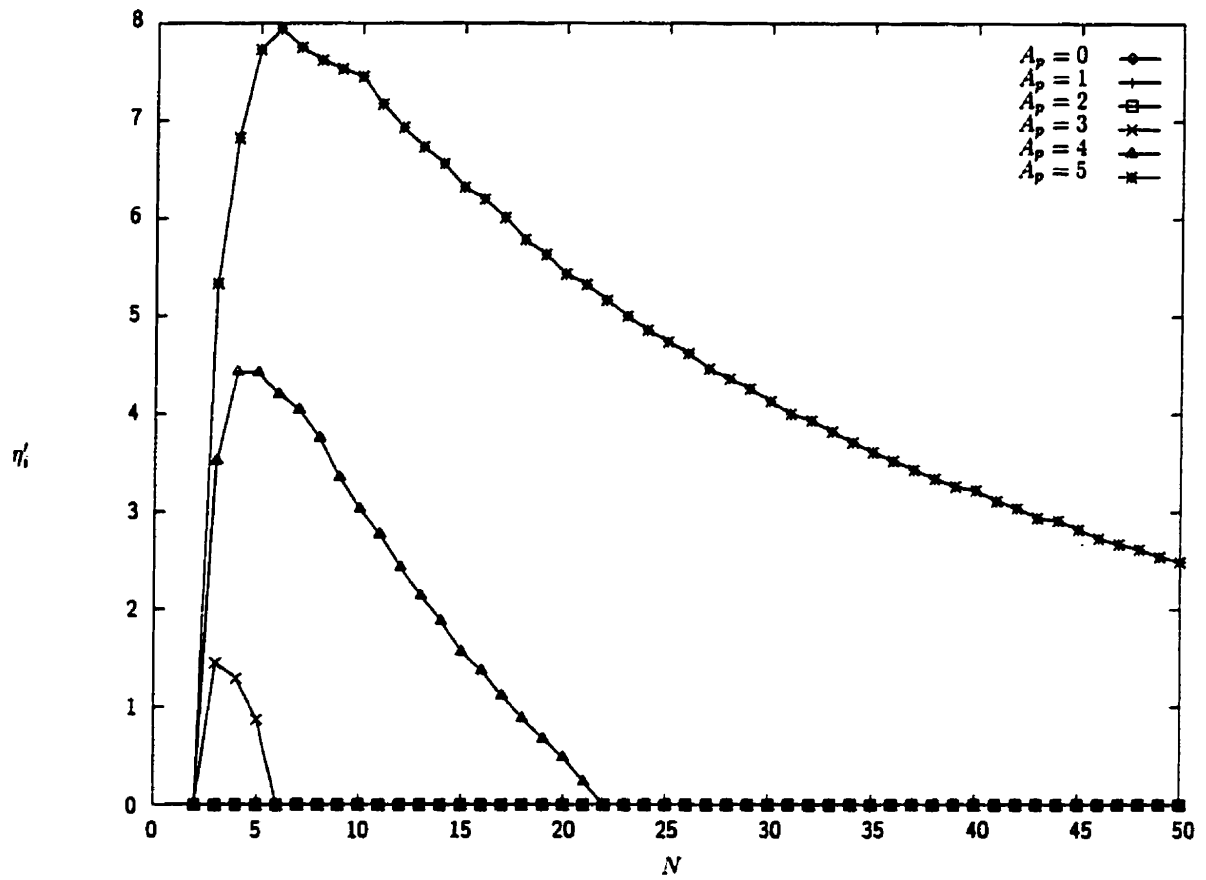


Figure 5.16: η'_i (en %) en fonction de N avec du point-à-point.

faire. Cependant, si les nœuds participant aux communications multipoint varient, on n'a pas le choix que de modifier les arbres faute de quoi, certains nœuds ne seront pas connectés.

2. Une deuxième possibilité est de ré-optimiser l'acheminement des appels sur d'autres arbres qu'il faudra éventuellement générer. Cette approche a l'avantage d'être simple et de pouvoir se faire sur une échelle de temps assez courte (quelques minutes). Cependant, elle peut être insuffisante si la variation de trafic est importante. De plus, elle exige une connectivité assez forte dans le réseau ce qui n'est pas le cas du réseau que nous obtenons en concentrant le trafic. Nous verrons dans la section 5.6, ce qu'il y a lieu de faire afin de faire en sorte que le réseau soit suffisamment connecté sans que cela ne coûte trop cher.
3. L'opérateur du réseau peut optimiser le dimensionnement des capacités des arcs sans pour autant ré-optimiser l'acheminement des appels. Ceci peut se faire si le réseau de transmission est un réseau logique par exemple, un réseau de VPs en ATM. Aussi, si les destinations changent, il faudra d'abord mettre à jour les arbres car autrement, certaines nouvelles destinations (commutateurs) pourraient ne pas être connectées alors que d'autres nœuds ayant quitté les communications pourraient recevoir inutilement du trafic.
4. Enfin, une dernière façon de réagir à la variation de trafic, qui donne les meilleurs résultats mais, qui ne peut être faite que sur une échelle de temps plus longue (plusieurs semaines) est de refaire la synthèse du réseau pour tenir compte des nouvelles valeurs des demandes de trafic. On peut alors utiliser

l'algorithme de concentration de trafic. Cette approche ne peut se faire fréquemment lorsqu'on dimensionne des liens physiques. Par contre, dans le cas où on a des liens logiques comme les VP [41] dans les réseaux de type ATM, le re-dimensionnement du réseau pourrait se faire assez fréquemment (ce type de réseau peut être re-configuré en moins d'une seconde aujourd'hui) (bien entendu, on ne va pas re-configurer le réseau toute les secondes).

Exemple 5.5.1 Afin d'étudier l'impact de la variation du trafic sur la performance du réseau, nous considérons le cas réaliste où on a suffisamment de trafic c.-à-d., un nombre suffisamment grand de types d'appel point-à-point et point-à-multipoint.

Soit les données des tableaux 5.18 et 5.19 où nous donnons les demandes de trafic et les destinations, prévues et réelles, correspondant aux types d'appel point-à-multipoint et point-à-point, respectivement. Nous avons supposé que l'on ne commet pas d'erreur sur les paires origine-destination (o-d) participant aux appels point-à-point.

Nous générons un réseau optimal à l'aide de l'algorithme de concentration de trafic (globale et individuelle) pour le trafic prévu (demandes de trafic et destination) et nous évaluons la performance de ce réseau lorsque le trafic varie, tel qu'indiqué dans les tableaux 5.18 et 5.19, en fonction de l'action entreprise par l'opérateur du réseau.

Les résultats sont donnés dans le tableau 5.20 pour le cas des appels point-à-multipoint exclusivement et dans le tableau 5.21 pour le cas où on a aussi des appels point-à-point. Dans le premier cas, dans la mesure où le réseau est un arbre, on ne peut ré-optimiser l'acheminement et par conséquent, on a un seul choix d'arbre pour connecter les appels si le trafic change. Cependant, lorsque les destinations

Tableau 5.18: *Demandes de trafic point-à-multipoint et destinations prévues et réelles. Les destinations prévues et qui ne le devraient pas sont en petit caractères et les destinations réelles non prévues sont en caractères gras.*

k	source	destinations	A^k Prévu	A^k Réel
1	1	{2,4,5,6,7,8}	4	2
2	2	{1,4,5,6,8,9}	3	2
3	3	{1,2,4,6,7,8,9}	4	5
4	4	{1,2,3,5,6,8,9}	5	4
5	5	{1,2,3,4,7,8,9}	5	7
6	6	{3,4,5,8,9}	2	4
7	7	{1,2,4,5,6,8,9}	6	6
8	8	{1,2,3,4,6,9}	5	4
9	9	{1,2,3,5,8}	3	2
10	5	{1,2,4,6,7,9}	1	2

Tableau 5.19: *Demandes de trafic point-à-point prévues et réelles.*

k	11	12	13	14	15	16	17	18	19	20
(o,d)	(7,3)	(2,4)	(1,6)	(8,9)	(4,7)	(6,1)	(2,8)	(3,9)	(3,4)	(2,1)
A^k Prévu	5	4	2	4	3	4	5	3	5	4
A^k Réel	4	5	3	2	5	3	4	4	4	5

changent, il faut modifier les arbres faute de quoi, certaines destinations pourraient ne pas être connectées. Dans le deuxième cas, puisque le réseau n'est pas un arbre, on peut ré-optimiser l'acheminement des appels si le trafic varie. L'optimisation de l'acheminement est réalisée par la méthode heuristique décrite dans le chapitre 4.

Tableau 5.20: *Performance du réseau en présence de variation de trafic avec exclusivement du trafic point-à-multipoint.*

Trafic	Action	Coût	Revenu	Valeur nette	$\max_k L^k$
A^k prévus	Aucune	34.8	218.5	183.7	.007
A^k réels	Aucune	34.8	218.0	183.2	.008
	Dimensionnement	35.0	218.5	183.5	.008
	RAQO	35.0	218.5	183.5	.008
Dest. réelles	Modif. arbres	34.8	214.2	179.4	.032
	Dimensionnement	35.6	218.5	182.9	.008
	RAQO	34.7	218.5	183.8	.008
A^k et dest. réels	Modif. arbres	34.8	214.3	179.5	.036
	Dimensionnement	36.7	224.5	187.2	.008
	RAQO	36.1	224.4	187.7	.008

Comme on peut le voir dans le tableau 5.20, même si aucune action n'est prise, la performance du réseau est peu sensible à la variation des demandes de trafic. Ainsi, on peut dire que dans le cas où on a exclusivement des appels point-à-multipoint, la connaissance exacte des demandes de trafic importe peu (à condition que la variation ne soit pas trop forte).

On peut aussi voir dans le tableau 5.21 que les erreurs de prédiction affectent

Tableau 5.21: *Performance du réseau en présence de variation de trafic avec du trafic point-à-multipoint et point-à-point. $k \leq 10$ correspond au appels point-à-multipoint.*

Trafic	Action	Coût	Revenu	valeur nette	$\max_{k \leq 10} L^k$	$\max_{k \geq 11} L^k$
A^k Prévus	Aucune	42.2	257.2	215.0	.008	.010
A^k réels	Aucune	42.2	251.5	209.3	.048	.034
	Opt. ach.	42.2	254.8	212.6	.022	.011
	Dim.	43.7	257.2	213.5	.008	.010
	Concentration	42.0	257.1	215.1	.010	.010
Dest. réelles	Opt. ach.	42.2	254.2	212.0	.025	.015
	Dim.	42.0	257.2	215.2	.009	.010
	Concentration	41.4	257.2	215.8	.009	.010
A^k et dest. réels	Opt. ach.	42.2	252.8	210.6	.040	.031
	Dim.	43.4	263.2	219.8	.008	.010
	Concentration	43.2	263.2	220.0	.008	.010

surtout le trafic point-à-multipoint. En effet, la perte d'appel est beaucoup plus importante pour ces types d'appel. Aussi, si aucune action n'est prise, les erreurs de prédiction sur les demandes de trafic ont un impact important et ceci est surtout dû à l'erreur supplémentaire sur les demandes de trafic point-à-point. Néanmoins, on peut améliorer la performance du réseau simplement en ré-acheminant les appels.

D'autre part, d'après les résultats du tableau 5.20, on constate que l'erreur de prévision sur les destinations a un impact non négligeable sur les performances du réseau. Il faut au moins re-dimensionner les arcs pour y remédier. Comme on peut le voir dans le tableau 5.21, la présence d'appels point-à-point permet de réduire l'impact des erreurs de prévision sur les destinations car dans ce cas, le réseau est relativement fortement connecté faisant en sorte qu'il est possible de ré-acheminer les appels pour améliorer la performance du réseau.

D'autre part, nous avons constaté (voir tableaux 5.20 and 5.21) que le coût du réseau obtenu après re-dimensionnement des artères de transmission est légèrement supérieur à celui obtenu en utilisant l'algorithme de concentration de trafic [71], et que les deux réseaux ont pratiquement les mêmes revenus et QS, ce qui laisse suggérer que *pourvu que le trafic soit concentré sur un petit nombre d'arcs susceptibles de servir à connecter un nombre assez grand de paire de nœuds participant aux différents types d'appel, peu importe ces arcs, le coût du réseau est quasi-optimal*, un résultat typique obtenu pour plusieurs exemples numériques.

5.6 Dilemme acheminement/dimensionnement

Nous avons illustré comment la concentration de trafic sur un petit nombre d'arcs réduisait considérablement le coût du réseau. D'autre part, dans le chapitre 4, nous

avons illustré comment la répartition du trafic sur des arbres disjoints augmentait le revenu généré par les appels. Nous nous demandons alors que faire en pratique si le réseau dimensionné ne comprend qu'un petit nombre d'arcs, donnant un choix très limité pour générer des arbres disjoints, alors que nous voulons ré-acheminer les appels quand le trafic varie.

Heureusement, en pratique, si nous avons des appels point-à-point et des appels point-à-multipoint et que ceux-ci ont les mêmes exigences en termes de largeur de bande, nous savons que la topologie optimale est plus connectée que celle d'un arbre et si nous avons assez d'appels point-à-point dans le réseau (ce qui est le cas en pratique) nous aurons une topologie qui permet de ré-optimiser l'acheminement des appels. Ainsi, le dilemme acheminement/dimensionnement n'est pertinent que s'il n'y a pas de trafic point-à-point dans le réseau ou si celui-ci requiert une petite largeur de bande.

Dans ce cas, pour rendre l'optimisation de l'acheminement des appels possible, nous avons besoin d'introduire plus de choix d'arcs dans le réseau. Une possibilité serait de modifier le modèle de sélection d'arbres, dans le contexte de la synthèse de réseaux, en introduisant une contrainte sur la connectivité minimale du réseau mais ceci augmente considérablement la complexité du problème. Une autre possibilité est de sélectionner des arbres disjoints en considérant tous les types d'appels ce qui est totalement à l'encontre du principe de concentration de trafic. En effet, nous avons montré qu'une telle solution donne un réseau dont le coût peut être 45% plus élevé qu'une valeur presque optimale que l'on peut atteindre [71] (voir exemple 5.4.5).

Néanmoins, nous pouvons générer un réseau relativement connecté de façon plus efficace en n'introduisant que des arcs pertinents à l'acheminement des appels. Ces

arcs sont ceux qui peuvent être partagés par le plus grand nombre possible de types d'appel tel que décrit dans la section 5.4.4 où nous avons dit que les arcs en question sont ceux ayant les plus fortes valeurs des ℓ_s .

Soit $h(n)$, le degré d'un nœud n . Si nous voulons que l'optimisation de l'acheminement soit pertinente, nous voulons éviter d'avoir plusieurs nœuds avec $h(n) = 1$ ce qui est le cas de toutes les feuilles d'un arbre. L'idée est alors, une fois que nous avons généré un RAQO par l'algorithme de la section 5.4.4 (concentration globale du trafic), nous continuons la sélection des arcs connectant les nœuds ayant à la fois des petites valeurs de $h(n)$ et des grandes valeurs de ℓ_s . En faisant ainsi, nous sommes sûrs que les arcs sélectionnés pourront servir à écouler beaucoup de trafic, provenant de plusieurs types d'appels et que nous allons efficacement augmenter la connectivité du réseau. Ceci peut être réalisé par la procédure suivante:

1. $h_0 = 1$.
2. Marquer les arcs correspondant aux paires de nœuds (i, j) telles que $h(i) = h_0$ et $h(j) = h_0$. S'il n'y a pas de tels arcs, aller à 3. Sinon, aller à 4.
3. Marquer les arcs correspondant aux paires de nœuds (i, j) telles que $h(i) = h_0$ ou $h(j) = h_0$. S'il n'y a pas de tels arcs, aller à 5.
4. Sélectionner l'arc (a, b) marqué et ayant la plus forte valeur de ℓ_s , et l'ajouter au réseau s'il n'y est pas déjà. $h(a) \leftarrow h(a) + 1$ et $h(b) \leftarrow h(b) + 1$. Enlever le marquage de tous les arcs et aller à 2.
5. $h_0 \leftarrow h_0 + 1$, si $h_0 > h_{\min}$ arrêt, sinon, aller à 2.

où h_{\min} est le degré minimal requis pour chaque nœud du réseau qui est une donnée

de l'algorithme (que l'on peut fixer selon la connectivité voulue du réseau) et h_0 est une variable entière qui représente le degré courant des nœuds.

Il reste cependant un problème à résoudre, quelles capacités allouer aux arcs sélectionnés, y compris ceux de l'arbre (RAQO) généré par l'algorithme de la section 5.4.4. Nous savons que nous voulons d'un côté minimiser le coût du réseau et de l'autre côté, obtenir une topologie assez connectée. Ainsi, on doit éviter de sélectionner des arbres utilisant exclusivement les arcs du RAQO car si on le faisait, au dimensionnement optimal, seuls les arcs appartenant à cet arbre auront une capacité positive. On doit plutôt sélectionner des arbres disjoints. La sélection d'arbres disjoints dans un réseau donné peut alors être réalisée à l'aide de l'heuristique de sélection d'arbres que nous avons proposée dans le chapitre 4. Maintenant que nous avons sélectionné les arbres, nous optimisons les capacités des arcs à l'aide de l'algorithme décrit dans la section 3.4.2.

Exemple 5.6.1 Dans cet exemple, nous considérons des appel point-à-multipoint exclusivement. Considérons les données correspondant aux demandes de trafic et destinations prévues données dans le tableau 5.18. Le RAQO correspondant est donné dans la figure 5.17 (a). Le coût de ce réseau est 34.8 pour $C_0 = 0.1$. On peut voir sur cette figure que $h(1) = 5$ et $h(n) = 1 \quad \forall n \in \{3, 5, 6, 7, 9\}$.

Maintenant, nous fixons $h_{min} = 2$ et nous utilisons l'algorithme de la section 5.6 pour augmenter la connectivité du réseau et nous obtenons alors le réseau de la figure 5.17 (b). Comme on peut le voir sur cette figure, trois arcs ont été ajoutés au RAQO, les arcs (5, 9), (3, 6) et (4, 7) que nous donnons par ordre décroissant des ℓ_j . Le coût de ce réseau est 38.5 ce qui est 11% plus cher que le RAQO de la figure 5.17 (a).

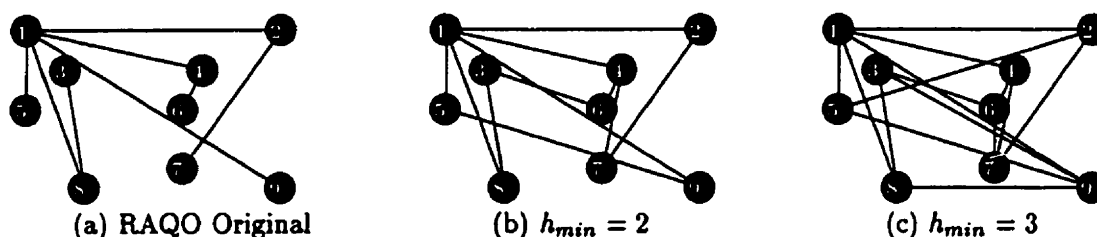


Figure 5.17: *Augmentation de la connectivité du réseau.*

On fait de même pour $h_{min} = 3$ et nous avons constaté que quatre arcs supplémentaires sont ajoutés au réseau précédent, les arcs (8, 9), (2, 5), (6, 7) et (3, 9) donnant lieu au réseau de la figure 5.17 (c). Le coût de ce réseau est 41.8 ce qui est 20% plus cher que le réseau de la figure 5.17 (a).

Afin de tester la performance des ces deux réseaux, nous considérons le cas où les demandes de trafic ainsi que les destinations varient en même temps, tel que indiqué dans le tableau 5.18. Nous considérons alors cinq cas: le cas où nous modifions les arbres dans le réseau de la figure 5.17 (a), deux cas où nous augmentons la connectivité du réseau et ensuite on utilise l'heuristique du chapitre 4 pour sélectionner de nouveaux arbres, un cas où nous dimensionnons les artères de transmission (pour les arbres modifiés obtenus dans le premier cas) et enfin, le cas où nous générons un RAQO pour le nouveau trafic. Les résultats sont donné dans le tableau 5.22.

Comme on peut le voir d'après ces résultats, en augmentant la connectivité du réseau, nous pouvons augmenter sensiblement le revenu et la QS en ré-optimisant l'acheminement des appels. Cependant, il faut avoir $h_{min} = 3$ au moins afin d'obtenir de bonnes performances pour le réseau lorsque le trafic varie. Le réseau obtenu pour $h_{min} = 3$ est 15% plus cher que le RAQO obtenu par concentration du trafic alors que le revenu est 5% plus élevé que celui qu'on aurait si on se contentait de modifier

Tableau 5.22: *Performance en présence de variation de trafic.*

	Coût	Revenu	Valeur nette	$\max_k L^k$
Modification des arbres	34.8	214.3	179.5	.036
3 arcs de plus ($h_{\min} = 2$)	38.5	222.1	183.6	.021
7 arcs de plus ($h_{\min} = 3$)	41.8	224.3	182.5	.009
Dimensionnement (trafic réel)	36.7	224.5	187.2	.008
RAQO (trafic réel)	36.1	224.4	187.7	.008

les arbres. Ceci nous indique que *si nous voulons augmenter la connectivité du réseau afin de pouvoir ré-optimiser l'acheminement des appels en cas de variation de trafic, le prix à payer est beaucoup plus important que le gain que nous obtenons en ré-optimisant l'acheminement. On a plutôt intérêt à refaire la synthèse optimale pour la nouvelle matrice de trafic*, un résultat typique qui a été observé pour plusieurs exemples. Notons que ce résultat est indépendant des coûts de capacité C_0 et revenus $w^k(c)$ (car nous considérons les pourcentages des accroissements du coût du réseau et du revenu). Il faudra éventuellement traduire C_0 et $w^k(c)$ en dollars afin d'avoir une idée sur ce que cela coûte en réalité d'augmenter la connectivité du réseau d'une part et sur le revenu généré par la ré-optimisation de l'acheminement des appels, d'autre part. De plus, nous n'avons pas tenu compte du coût rattaché à l'opération qui consiste à ajouter des arcs au réseau.

Exemple 5.6.2 Dans cet exemple, nous considérons des réseaux écoulant à la fois du trafic point-à-point et du trafic point-à-multipoint. Les résultats sont donnés dans le tableau 5.23 où nous donnons le nombre d'arcs dans le réseau en fonction

du nombre de nœuds et de ρ , le rapport du nombre de types d'appel point-à-point sur le nombre de types d'appel point-à-multipoint. Comme on peut le voir dans ce tableau, le nombre d'arcs est suffisamment grand faisant en sorte que si le trafic varie, on peut ré-acheminer les appels en conséquences et améliorer la performance du réseau.

Des résultats pour $N = 9$ sont donnés dans le tableau 5.21 de l'exemple 5.5.1. On peut remarquer que même si nous avons seulement 11 arcs dans le réseau, on peut améliorer sa performance en ré-optimisant l'acheminement en cas de variation de trafic, surtout dans le cas où seulement les intensités des demandes de trafic ou les destinations varient. Il va sans dire que si le trafic augmente considérablement, il faut re-dimensionner le réseau comme on peut le voir dans le tableau 5.21 dans le cas où les intensités des demandes de trafic ainsi que les destinations varient.

Tableau 5.23: *Nombre d'arcs dans le réseau vs N et ρ .*

N	$\rho = 15/30$	$\rho = 20/40$	$\rho = 20/30$	$\rho = 26/39$
9	16	19	17	22
16	35	40	36	38
25	34	40	36	40

5.7 Utilité pratique de l'heuristique de sélection d'arbres

De même que pour l'acheminement, l'heuristique que nous avons proposée pour la sélection d'arbre dans le contexte de la synthèse de réseaux dépend de la connaissance à priori des matrices de trafic. Or, en pratique, il peut être difficile de caractériser ces matrices de trafic et nous pouvons avoir de grosses différences entre les valeurs prévues et les valeurs réelles de ces matrices de trafic. On se demande alors que faire si en pratique, nous ne connaissons pas ces matrices de trafic et que nous voulons dimensionner le réseau.

Si on constate que même avec une bonne sélection d'arbres (obtenue par réacheminement des appels par la méthode heuristique décrite dans le chapitre 4) les performances du réseau ne sont pas acceptables, l'opérateur du réseau n'a pas le choix que de re-dimensionner les artères de transmission. Alors, il a deux choix, soit garder la sélection d'arbre et optimiser les capacités en utilisant l'algorithme de la section 3.4.2 (dimensionnement optimal) ou encore, refaire la synthèse du réseau en tenant compte des nouvelles demandes de trafic (mesurées) et en re-sélectionnant les arbres par l'algorithme de concentration de trafic.

La mise à jour des capacités des artères de transmission peut se faire approximativement en utilisant la relation (5.7), en augmentant ou en diminuant la capacité de l'arc si la probabilité de blocage mesurée est inférieure ou supérieure à celle donnée par l'équation (5.6), respectivement. La concentration de trafic peut être réalisée en mesurant le trafic offert aux arcs et en utilisant ces mesures pour approximer les ℓ_j .

C'est à l'opérateur du réseau de décider quand il est nécessaire de mettre à

jour les capacités des arcs où encore, quand il doit refaire la synthèse du réseau en re-sélectionnant les arbres. Nous avons montré qu'en général, il suffit de mettre à jour les capacités dans la mesure où le coût du réseau obtenu après mise à jour des capacités des liens est légèrement plus cher que le réseau obtenu en re-sélectionnant les arbres.

La fréquence à laquelle on refait le dimensionnement ou la synthèse optimale du réseau dépend de la nature des liens c.-à-d., si les liens sont logiques, on peut le faire beaucoup plus fréquemment que dans le cas de liens physiques. Typiquement, toutes les semaines pour des liens logiques et quelque mois pour des liens physiques. Cependant, pour les réseaux de type ATM, on parle aujourd'hui de la possibilité de re-configurer le réseau en moins d'une seconde.

5.8 Autres méthodes d'optimisation de la sélection d'arbres

Dans cette section, nous comparons l'heuristique de concentration de trafic à ce qu'on aurait si on avait utilisé des algorithmes génétiques ou des recherche du type tabou (voir section 4.5).

5.8.1 Algorithmes génétiques

Dans le contexte de la synthèse de réseaux, le croisement entre deux arbres de types d'appels différents donnerait lieu à un arbre qui utilise le maximum possible d'arcs en commun avec eux. Cependant, dans la mesure où dans le contexte de synthèse on veut minimiser le nombre d'arcs et d'arbres, le croisement est inapproprié

(on choisirait une probabilité de croisement nulle). La mutation consisterait à altérer un arbre donné afin de minimiser le nombre d'arcs qu'il utilise et c'est ce que nous faisons pour les types d'appel ayant un petit nombre de destinations dans la section 5.4.5. Ainsi, on peut voir que cette approche est à peu de choses près semblable à la méthode heuristique que nous proposons dans la section 5.4.3.

5.8.2 Recherche de type tabou

On peut supposer qu'un voisinage est constitué de l'ensemble de réseaux qui diffèrent de i arcs du réseau actuel. On a alors deux choix, soit remplacer i arcs ou ajouter i arcs.

Pour chaque réseau dans le voisinage, il faudra sélectionner les arbres pour chaque type d'appel. Par exemple, dans le cas où on ajoute un seul arc au réseau, on a N arcs pour N nœuds. Il y a alors plus d'une façon de générer un arbre pour certains types d'appel. Ensuite, il faudra déterminer les capacités optimales des arcs ainsi que le partage de charge. Dans chaque itération, nous retenons le réseau qui a la meilleure valeur nette. On peut montrer à l'aide d'exemples numériques que même avec une aussi petite altération du réseau, cela prend un temps très long pour explorer tout le voisinage.

Nous avons appliqué cette méthode en partant du RAQO, dans le cas d'appels point-à-multipoint exclusivement, tel que décrit dans la section 5.4.4. Dans le cas où nous limitons le voisinage aux réseaux en arbre qui diffèrent d'un seul arc du RAQO, les résultats sont donnés dans le tableau 5.24. On peut voir que les réseaux en arbre dans le voisinage sont entre 0% et 14% plus chers que le RAQO alors que le revenu et la QS sont pratiquement les mêmes [71]. On peut aussi voir que lorsqu'on

remplace l'arc s_1 par l'arc s_2 , le coût du réseau qui en résulte augmente avec $\ell_{s_1} - \ell_{s_2}$ ce qui illustre l'optimalité de la concentration du trafic. Notons que nous pouvons avoir plusieurs RAQO avec la concentration globale du trafic et le meilleur réseau donné dans le tableau 5.24 n'en est pas un.

Tableau 5.24: *Performance du réseau quand l'arc s_1 est remplacé par l'arc s_2 .*

Réseau	$s_1 : (i, j)$	$s_2 : (i, j)$	$\ell_{s_1} - \ell_{s_2}$	Coût	Revenu	Val. nette	$\max_k L^k$
pire	1:(1,2)	19:(3,7)	31	39.5	218.4	178.9	.008
arbitraire	4:(1,5)	17:(3,5)	17	37.7	218.4	180.7	.009
meilleur	20:(3,8)	2:(1,3)	2	34.8	218.5	183.7	.008
RAQO	–	–	–	34.8	218.5	183.7	.007

5.8.3 Résumé

Nous avons brièvement expliqué comment on pourrait sélectionner les arbres en utilisant des méthodes génétiques ou de recherche de type tabou. Nous avons illustré les ressemblances qui existent entre ces méthodes et notre heuristique et nous avons expliqué pourquoi notre méthode est plus appropriée pour résoudre le problème de sélection d'arbres.

5.9 Conclusion

Nous avons illustré dans quelle mesure la concentration du trafic permet de réduire le coût du réseau. Nous avons montré que la concentration de trafic est le

critère selon lequel les arbres doivent être sélectionnés dans le contexte de la synthèse de réseaux. Nous avons aussi montré que ce résultat est indépendant des coûts de capacité.

Nous avons décrit un algorithme basé sur une méthode exacte pour résoudre le problème de sélection d'arbres pour la synthèse de réseaux. Nous avons constaté qu'un tel algorithme est pratiquement inutilisable à cause de sa complexité. Nous avons alors proposé une heuristique simple basée sur le critère de sélection d'arbres pour résoudre ce problème. Pour tous les exemples étudiés, cette heuristique a donné des résultats très intéressants notamment, une économie de coût du réseau pouvant aller jusqu'à 45% par rapport au cas où nous sélectionnons les arbres de façon arbitraire, ce qui nous laisse croire que nous sommes très proches d'un optimum global.

Nous avons aussi montré qu'il y a un dilemme concernant le dimensionnement optimal du réseau pour du trafic point-à-point d'une part et pour du trafic point-à-multipoint d'autre part. Plus le trafic point-à-point est intense par rapport au trafic point-à-multipoint, plus la topologie optimale est connectée et vice-versa. Nous avons aussi montré que dans le cas où nous avons exclusivement des appels point-à-multipoint, nous avons un autre dilemme concernant le dimensionnement et l'acheminement cette fois. Le réseau optimal obtenu par dimensionnement est faiblement connecté se réduisant souvent à un arbre, ce qui ne permet pas de ré-optimiser l'acheminement des appels en cas de variation du trafic. Nous avons proposé une heuristique pour augmenter la connectivité du réseau de façon économique et nous avons constaté que le prix à payer afin d'être en mesure de ré-optimiser l'acheminement des appels est beaucoup plus important que le gain réalisé par ce ré-acheminement.

Enfin, nous avons constaté que la variation des demandes de trafic avait un impact très faible sur les performances du réseau alors que la variation des destinations avait un impact très important.

CHAPITRE 6

Conclusions

Dans ce chapitre, nous résumons les contributions effectuées dans le cadre de ce travail de recherche. Nous donnons aussi un aperçu sur d'autres problèmes portant sur les réseaux avec communications multipoint qui pourraient faire l'objet d'autres travaux de recherche.

6.1 Résumé des contributions

La première contribution consiste à avoir clarifié le problème de synthèse de réseaux avec communications multipoint et à avoir identifié les points essentiels qu'il faut étudier.

Nous avons développé des outils informatiques qui permettent de faire diverses études portant sur le problème de synthèse de réseaux avec communications multipoint notamment, un algorithme d'optimisation pour l'acheminement des appels et un algorithme de dimensionnement optimal du réseau.

À l'aide des algorithmes d'optimisation, nous avons illustré l'impact de la sélection d'arbres sur la synthèse des réseaux de télécommunications. Nous avons ensuite établi deux critères simples et efficaces de sélection d'arbres, un dans le contexte d'acheminement et un dans le contexte de synthèse de réseaux. Ceci a fait l'objet d'une publication dans une conférence internationale [70].

Nous avons ensuite modélisé le problème de sélection d'arbres pour les problèmes d'acheminement et de dimensionnement. Pour le problème d'acheminement, nous avons développé un algorithme basé sur la méthode du gradient projeté et sur la méthode de génération de colonnes pour résoudre le problème de sélection d'arbres de façon exacte et nous avons montré les limitations de cette approche. Nous avons ensuite proposé une méthode heuristique plus rapide et de qualité comparable à la méthode exacte. Ce travail a été soumis pour publication dans une revue spécialisée [72] et l'heuristique a fait l'objet de l'article de conférence [74]. Nous avons montré comment on pourrait résoudre le problème de sélection d'arbres dans le contexte de la synthèse de réseaux de façon exacte et nous avons illustré la complexité d'une telle approche. Nous avons alors proposé une méthode heuristique pour résoudre le problème de façon approximative. Ce travail a aussi été soumis pour publication à une revue spécialisée [71] et l'heuristique a fait l'objet de l'article de conférence [73].

6.1.1 Sélection d'arbres

Nous avons présenté deux algorithmes pour la sélection optimale d'arbres dans le contexte l'acheminement. Le premier algorithme converge théoriquement vers un optimum local mais il est très complexe et la qualité de ses solutions dépend considérablement de la solution initiale. Le deuxième algorithme est basé sur une méthode

heuristique et nous permet de trouver rapidement une bonne sélection d'arbres qui peut servir entre autre comme solution initiale pour la méthode exacte. Néanmoins, pour certains exemples, les solutions obtenues par la méthode heuristique sont optimales pour la méthode exacte.

Nous avons aussi décrit un algorithme basé sur une méthode exacte pour résoudre le problème de sélection d'arbres pour la synthèse de réseaux et nous avons constaté qu'un tel algorithme est pratiquement inutilisable. Nous avons alors proposé un algorithme basé sur une heuristique simple pour résoudre ce problème approximativement. Pour tous les exemples étudiés, cet algorithme a donné des résultats très encourageants notamment une économie de coût du réseau pouvant aller jusqu'à 45% par rapport au cas où nous sélectionnons les arbres de façon arbitraire.

À l'aide de ces algorithmes, nous avons illustré dans quelles mesures l'utilisation d'arbres disjoints augmentait le revenu des connexions d'une part et dans quelle mesure la concentration du trafic permet de réduire le coût du réseau d'autre part. Nous avons aussi illustré la complexité du problème de sélection d'arbres et nous avons montré qu'il n'est pas convexe.

6.1.2 Variation du trafic

Nous avons étudié le problème de sensibilité du réseau obtenu par synthèse optimale à la variation de trafic due à des erreurs de prédiction soit sur les demandes de trafic ou sur les ensembles de nœuds participants dans les communications point-à-multipoint.

Nous avons constaté que dans le contexte de communications point-à-multipoint, une erreur de prédiction sur les demandes de trafic n'avait pas beaucoup d'effet sur

les performances de réseau alors qu'une erreur sur les membres d'une communication multipoint engendre une perte de revenu et une dégradation de la QS non négligeables.

Nous avons illustré comment l'opérateur du réseau peut réagir afin de pallier la dégradation de performance causée par la variation de trafic. Une façon de réagir à la variation de trafic est de garder le même réseau, avec les mêmes capacités, et ré-optimiser l'acheminement des appels sur d'autres arbres qu'il faudra éventuellement générer. Cette approche a l'avantage d'être simple et de pouvoir se faire sur une échelle de temps assez courte (quelques minutes). Cependant, elle peut être insuffisante si la variation de trafic est importante. Une autre façon, qui donne de meilleurs résultats mais, qui ne peut être faite que sur une échelle de temps plus longue (plusieurs jours) est de re-dimensionner le réseau en gardant la même sélection d'arbres pour tenir compte des nouvelles valeurs des demandes de trafic. Comme dernier recours, l'opérateur du réseau peut refaire la synthèse du réseau en sélectionnant de nouveaux arbres. Cependant, la synthèse du réseau ne peut se faire fréquemment lorsqu'on a des liens physiques. Par contre, dans le cas où on a des liens logiques comme les VP (Virtual Path) [41] dans le contexte de l'ATM, la synthèse du réseau pourrait se faire assez fréquemment.

6.1.3 Dilemme point-à-point/multipoint

Nous avons étudié le cas de réseaux écoulant à la fois du trafic point-à-point et point-à-multipoint. Nous avons constaté que la topologie du réseau varie selon

les intensités des deux types de trafic. On obtient alors des topologies très fortement connectées dans le cas où le trafic point-à-point est plus intense que le trafic multipoint, et des topologies faiblement connectées dans le cas contraire.

Nous avons aussi montré que puisque la présence du trafic point-à-point avec le multipoint fait en sorte que la topologie optimale du réseau est plus fortement connectée, d'une part, nous pouvons ré-optimiser l'acheminement des appels si le trafic varie et d'autre part, nous introduisons une certaine sécurité dans le réseau commuté.

6.1.4 Dilemme dimensionnement/acheminement

Nous avons illustré comment la concentration de trafic sur un petit nombre d'arcs réduisait considérablement le coût du réseau. D'autre part, nous avons illustré comment la répartition du trafic sur des arbres ayant le moins possible d'arcs en commun augmentait le revenu généré par les appels. On se demande alors que faire en pratique si le réseau dimensionné ne comprend qu'un petit nombre d'arcs donnant un choix très limité pour générer des arbres disjoints.

Si nous avons des appels point-à-point avec les appels point-à-multipoint, la topologie optimale est fortement connectée, ce qui rend possible la ré-optimisation de l'acheminement des appels. Ainsi, le dilemme acheminement / dimensionnement n'est pertinent que s'il n'y a pas de trafic point-à-point dans le réseau (ou si le trafic point-à-point a de faibles demandes de trafic et/ou requiert une petite largeur de bande).

Dans ce cas, pour rendre l'optimisation de l'acheminement des appels possible,

nous avons besoin d'introduire plus d'arcs dans le réseau. Nous avons montré comment générer un réseau relativement connecté de façon économique et ce, en n'introduisant que des arcs pertinents à l'optimisation de l'acheminement des appels. Ces arcs sont ceux qui peuvent être partagés par le plus grand nombre possible de types d'appel. Nous avons alors montré que le prix à payer afin de pouvoir ré-optimiser l'acheminement des appels en cas de variation de trafic est beaucoup plus important que le gain que nous obtenons en ré-optimisant l'acheminement.

6.1.5 Multi-débit

Nous avons étudié le problème de sélection d'arbres dans le contexte de réseaux multi-débit ce qui nous a permis entre autre d'appliquer nos résultats aux réseaux à large bande de type ATM en utilisant le concept de débit équivalent.

Nous avons montré que les conclusions obtenues dans le contexte du mono-débit sont aussi valables dans le contexte du multi-débit, notamment, la concentration de trafic dans le contexte de dimensionnement et la dispersion du trafic dans le contexte d'acheminement. De plus, nous avons montré que la concentration du trafic est encore plus avantageuse dans le contexte du multi-débit à cause des grandes largeurs de bande requises dans ce contexte.

6.2 Autres problèmes pertinents

Dans cette section, nous donnons plusieurs problèmes dont nous n'avons pas tenu compte dans nos modèles ou que nous avons simplifiés. Chaque problème soulevé ici pourrait faire l'objet d'un travail de recherche à part entière.

Dans notre étude, nous avons supposé que l'acheminement des appels était un

simple partage de charge. En pratique, on utilise d'autres techniques d'acheminement notamment les acheminements adaptatifs ou le débordement [40]. Nous devons donc vérifier la validité de la plupart de nos conclusions pour d'autres techniques d'acheminement. Néanmoins, on pourrait s'attendre à ce que dans le contexte de synthèse de réseaux, dans la mesure où un arbre unique est optimal pour acheminer les appels, les résultats obtenus soient indépendants de la technique d'acheminement [71]. Cependant, ceci n'est pas le cas pour des appels point-à-point exclusivement où on sait qu'avoir plusieurs chemins de débordement améliore la performance du réseau [38].

Aussi, nous nous sommes intéressés uniquement au cas des appels point-à-multipoint. Il y a plusieurs autres scénarios possibles notamment des communication multipoint-à-multipoint. Pour l'instant, l'ATM Forum propose de faire de telles communications moyennant plusieurs communications point-à-multipoint, autant qu'il en faut pour que tous les nœuds puissent communiquer entre eux [7]. Ceci n'est certainement pas la meilleure façon de faire si on considère par exemple le gaspillage de ressources en termes d'espace d'adressage. On devrait plutôt utiliser des arbres partagés et on doit alors modifier le modèle de synthèse de réseaux.

D'autre part, dans le modèle utilisé, nous avons supposé que pour un type d'appel donné, on a une source et que les destinations (commutateurs) sont fixes et connues d'avance. Dans le cas général, on a des communications multipoint-à-multipoint et les usagers peuvent joindre ou quitter une communication de façon dynamique ce qui peut affecter la composition de l'ensemble de commutateurs pour un type d'appel donné surtout lorsqu'il y a au plus un usager connecté par commutateur. Un

exemple typique serait une conférence ou on a un très petit nombre d'utilisateurs géographiquement éloignés. Cela changera considérablement le modèle et les résultats obtenus ici.

Nous avons aussi supposé que nous connaissions les matrices de trafic. Or, en pratique, ça ne sera probablement pas le cas. Le problème de connaissance de matrices de trafic ne concerne pas uniquement les communications multipoint mais les réseaux de communications en général. Néanmoins, nous avons constaté que la synthèse optimale pour des communications multipoint donnait un réseau peu sensible aux variations des demandes de trafic [71], ce qui voudrait dire que la connaissance des valeurs de ces demandes de trafic importe peu dans ce cas particulier (dans la mesure où les erreurs de prédiction ne sont pas trop fortes). D'autre part, nous avons considéré que les matrices de trafic étaient caractérisées par une valeur qui donne le taux d'arrivée Poissonien d'appels d'une source vers un certain nombre de destinations. Or, en pratique on peut avoir des communications multipoint-à-multipoint avec un très petit nombre d'utilisateurs qui ne se connectent pas nécessairement en même temps. Dans ce cas, la caractérisation des matrices de trafic que nous proposons n'est plus suffisante et il faudrait tenir compte de l'aspect dynamique des connexions.

Nous avons aussi supposé que tous les commutateurs du réseau pouvaient faire la duplication d'unités de données. Dans Internet, cela n'est pas le cas dans la mesure où on utilise des serveurs dupicateurs qui sont les seuls éléments du réseau capables de dupliquer les datagrammes. Cela introduit une contrainte importante dans le problème de génération d'arbres. Il faudrait alors étudier l'impact de l'utilisation de

tels serveurs sur la topologie optimale du réseau ainsi que sur les performances en termes de coût de dimensionnement, revenu et qualité de service.

D'autre part, nous n'avons pas tenu compte dans notre modèle des performances du niveau paquet (cellules pour les réseaux ATM) en terme de qualité de service (qs) [41] spécifique au multipoint. Notons au passage que la qs reste à définir dans le contexte du multipoint car on ne sait pas clairement ce que cela veut dire pour un ensemble d'utilisateurs. La difficulté vient surtout de la diversité des terminaux que les utilisateurs utilisent. Par exemple, on ne peut avoir les mêmes exigences de qs pour un terminal lent et un terminal très rapide qui prennent part à une même communication. Une quantité comme le délai de transmission de paquets de bout en bout, délai qui peut être augmenté à cause de l'opération de duplication de paquets, devrait faire partie du modèle. Il est possible de simplifier la modélisation en introduisant une telle quantité dans le débit équivalent. Seule une étude approfondie, en analysant les performances du niveau paquet pourrait dire si une telle simplification serait justifiable. Nous n'avons pas tenu compte non plus de la congestion que pourrait causer le trafic multipoint dans le réseau à cause de la corrélation due à la transmission instantanée d'un grand nombre de copies d'une même cellule à un point donné du réseau. Ce genre de trafic peut causer des congestions très importantes surtout si des serveurs dupliques sont utilisés.

Nous avons aussi considérablement simplifié l'aspect tarification dans notre modèle dans la mesure où nous avons supposé que chaque appel multipoint générerait un revenu. Ce revenu pourrait être interprété comme le prix total que devront payer ensemble, tous les utilisateurs participant à une communication multipoint donnée. Nous n'avons pas tenu compte de la contribution individuelle de ces utilisateurs de façon à ce

que le coût de la communication soit équitablement réparti parmi ces usagers. Dans un réseau réel, plus un usager est éloigné par rapport à la source, plus il est coûteux de le connecter et surtout de lui offrir une garantie de QS [46]. Il est ainsi tout à fait raisonnable de faire en sorte qu'un tel usager paye plus cher que les autres membres d'une communication multipoint qui sont plus proches que lui de la source.

D'autre part, le modèle de synthèse de réseau que nous utilisons exige une intervention de l'opérateur du réseau à chaque fois que le trafic change. Ce modèle pourrait être modifié de manière à optimiser la synthèse non pas pour une seule matrice de trafic, mais plutôt pour une plage de valeurs possible pour les demandes de trafic afin de tenir compte de la variation éventuelle de ces demandes de trafic. Une telle approche donnerait un réseau plus robuste, peu sensible à des variations de trafic mais certainement plus coûteux. Cependant, on ne sait pas quelle est la complexité qui serait introduite par une telle approche.

Nous avons aussi supposé que nous avons un modèle sans réservation de circuits. Cela nous a amené à conclure que dans ce cas, le trafic de faible débit a un impact négligeable sur la performance du réseau. Il faudra éventuellement tester cette conclusion qui ne sera plus vraie si on introduit la réservation de circuits dans le modèle. Un modèle avec réservation est susceptible d'être très complexe notamment, à cause de l'aspect dynamique des connexions puisqu'alors, il est très difficile de connaître le trafic à l'avance [77].

Enfin, dans le modèle de synthèse de réseaux, on a supposé que les arcs étaient symétriques. Un modèle plus réaliste doit tenir compte du fait qu'en pratique, les arcs sont asymétriques. En effet, dans un réseau de type ATM par exemple, un VC bidirectionnel est constitué de deux VC unidirectionnels de sens opposés. Cette

dissymétrie fait en sorte que les algorithmes de génération d'arbres classiques tels que ceux de Steiner doivent être modifiés [85] dans la mesure où on aura un graphe orienté et le coût d'un arc donné dépend du sens dans lequel cet arc est considéré; ce qui rendrait le problème de synthèse encore plus complexe qu'il ne l'est déjà.

RÉFÉRENCES

- [1] AHARONI, E. et COHEN, R. (1998). Restricted Dynamic Steiner Trees for Scalable Multicast in Datagram Networks. *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 6, no 3, pages 286–297.
- [2] ALMEROTH, K. C. et AMMAR, M. H. (1997). Multicast Group Behavior in the Internet's Multicast Backbone (Mbone). *IEEE COMMUNICATIONS MAGAZINE*, vol. 35, no 6, pages 124–129.
- [3] AMMAR, M. H., CHEUNG, S. Y., et SCOGGIO, C. M. (1993). Routing Multi-point Connections Using Virtual Paths in an ATM Network. Proc. *IEEE INFOCOM'93*, pages 98–105.
- [4] ARMITAGE, G. (1995a). Multicast and Multiprotocol support for ATM based Internets. Proc. *ACM SIGCOM'95*, pages 34–46.
- [5] ARMITAGE, G. (1995b). Support for Multicast over UNI 3.1 based ATM Networks. Rapport technique, Bellcore. draft-ietf-ipatm-ipmc-10.txt.
- [6] ARMITAGE, G. (1997). IP Multicasting over ATM Networks. *IEEE JSAC*, vol. 15, no 3, pages 445–457.

- [7] ATM Forum (1994). ATM User Network Interface Specification. Rapport technique, ATM Forum. Version 3.1.
- [8] ATM Forum (1996). ATM User Network Interface Signaling Specification. Rapport technique, ATM Forum. Version 4.0.
- [9] BALLARDY, T., FRANCIS, P., et CROWFORD, J. (1993). Core Based Trees (CBT) an Architecture for Scalable Multicast Routing. Proc. *SIGCOMM'93*.
- [10] BERTSEKAS, D. et GALLAGER, R. (1992). *Data Networks*. Prentice-Hall.
- [11] BILHARTZ, T., CAIN, J. B., FARREY-GOUDREAU, E., FIEG, D., et BATSELL, G. S. (1997). Performance and Resource Cost Comparisons for the CBT and PIM Multicast Routing Protocol. *IEEE JSAC*, vol. 15, no 3, pages 304–315.
- [12] BIRMAN, A., FIROIU, V., GUÉRIN, R., et KANDLUR, D. (1995). Provisioning of RSVP-based Services over Large ATM Networks. Rapport technique, IBM Research division.
- [13] BORDEN, M., CRAWLEY, E., DAVIS, B., et BATSELL, S. (1995). Integration of Real-time Services in an IP-ATM Network Architecture. Rapport technique, Network Working Group. RFC 1821.
- [14] BRADEN, R., ZHANG, L., BERSON, S., HERZOG, S., et JAMIN, S. (1997). Resource Reservation Protocol (RSVP). Rapport technique, Network Working Group. RFC 2205.
- [15] CALVERT, K. L., ZEGRA, E. W., et DONAHOO, M. J. (1995). Core Selection Methods for Multicast Routing. Proc. *IEEE GLOBECOM'95*, pages 638–642.

- [16] CERF, V. G. (1991). Guidelines for Internet Measurement Activities. Rapport technique, Network Working Group. RFC 1262.
- [17] CERF, V. G. (1996). Preface to this Special Issue on Computer Telephony. *IEEE COMMUNICATIONS MAGAZINE*, vol. 34, no 4, page 28.
- [18] CHUANG, S. (1995). A Flexible and Secure Multicast Architecture for ATM Networks. *Proc. ATM Document Collection 4 (The Green Book)*, chapter 13. University of Cambridge Computer Laboratory.
- [19] DALAL, Y. K. et METCALFE, R. M. (1978). Reserve Path Forwarding of Broadcast Packets. *COMMUNICATIONS OF THE ACM*, vol. 21, no 12.
- [20] DEERING, S. (1989). Host Extensions for IP Multicasting. Rapport technique, IETF. RFC 1112.
- [21] DEERING, S., ESTRIN, D., FARINACCI, D., JACOBSON, V., LIU, C., et WEI, L. (1996). The PIM Architecture for Wide-Area Multicast Routing. *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 4, no 2, pages 153–162.
- [22] DELMAIRE, H., LANGEVIN, A., et RIOPEL, D. (1996). A Decomposition Framework using Genetic Algorithms for Global Optimization. Rapport technique, Les cahiers du GERAD. G-96-08.
- [23] DIOT, C., DABBOUS, W., et CROWFORD, J. (1997). Multipoint Communications: A Survey of Protocols, Functions and Mechanisms. *IEEE JSAC*, vol. 15, no 3, pages 277–290.
- [24] DOAR, M. et LESLIE, I. (1993). How Bad is Naive Multicast Routing. *Proc. IEEE INFOCOM'93*, pages 82–89.

- [25] DZIONG, Z., LIAO, K., et MASON, L. G. (1993). Effective Bandwidth Allocation and Buffer Dimensioning in ATM Based Networks with priorities. *Computer Networks and ISDN-Systems*, vol. 25, pages 1065–1078.
- [26] EDDIE, K. L. et LEON-GARCIA, A. (1995). Multicast and Self-Routing in ATM Radix Trees and Banyan Network. Proc. *IEEE INFOCOM'95*, pages 951–959.
- [27] ELWALID, A. et MITRA, D. (1993). Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks. *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 1, no 3, pages 329–343.
- [28] ERIKSSON, H. (1994). MBONE: The Multicast Backbone. *COMMUNICATIONS OF THE ACM*, vol. 37, no 8, pages 54–60.
- [29] ESBENSEN, H. (1995). Computing Near Optimal Solutions for the Steiner Problem in a Graph using Genetic Algorithms. *Networks*, vol. 26, pages 173–185.
- [30] FAHMY, S., JAIN, R., KALY, S., GOYAL, R., VANDALORE, B., et CAI, X. (1997). A Survey of Protocols and Open Issues in ATM Multipoint Communications. Rapport technique, The Ohio State University. <http://www.cis.ohio-state.edu/jain/papers/mcast.htm>.
- [31] FENNER, W. (1997). Internet Group Management Protocol Version 2. Rapport technique, IETF. draft-ietf-idmr-igmp-v2-06.txt.
- [32] FONSECA, N. et ZUKERMAN, M. (1997). ATM Dimensioning and Traffic Management and Modeling. Proc. *IEEE GLOBECOM'97 Tutorials*.

- [33] FRANK, M. et WOLFE, P. (1956). An Algorithm for Quadratic Programming. *Naval Research Logistics Quarterly*, vol. 3, pages 95–110.
- [34] GAREY, M. et JOHNSON, D. (1979). *Computers and Intractability: A guide to the theory of NP-Completeness*. Freeman.
- [35] GAUTHIER, E., LE BOUDEC, J. Y., et OECHSLIN, P. (1997). SMART: A Many-to-Many Multicast Protocol for ATM. *IEEE JSAC*, vol. 15, no 3, pages 458–472.
- [36] GENDREAU, M., LAROCHELLE, J. F., et SANSÓ, B. (1999). A Tabu Search Heuristic for the Steiner Tree Problem. *à paraître dans Networks*.
- [37] GIRARD, A. Conception de réseaux de télécommunications. Notes de cours, INRS-Télécommunications.
- [38] GIRARD, A. (1990). *Routing and Dimensioning in Circuit-switched Networks*. Addison-Wesley.
- [39] GIRARD, A. et EL BOUAZAOU, G. (1993). An Optimization Method for Broadband ISDN Networks. Proc. *GLOBECOM'93*, pages 762–766.
- [40] GIRARD, A. et ROSENBERG, C. (1997). A Unified Model for Network Synthesis with Generalized Connections. Proc. *ITC-15*, pages 319–328.
- [41] GIRARD, A. et ROSENBERG, C. (1998). Dimensioning of ATM Networks with Finite Buffers under Call-Level and Cell-Level Constraints. *Broadband Communications*.
- [42] GLOVER, F. (1986). Future Paths for Integer Programming and link to Artificial Intelligence. *Computers & Operations Research*, vol. 13, pages 533–549.

- [43] GROSSGLAUSER, M. et RAMAKRISHNAN, K. K. (1997). SEAM: Scalable and Efficient ATM Multipoint-to-Multipoint Multicasting. *Proc. IEEE INFOCOM'97*, pages 867–875.
- [44] GUO-QING, H. (1998). Forest Build Algorithms for Multiple Destinations. *IEEE Potentials*, pages 13–16. Août/Septembre.
- [45] HANDLEY, M. et JACOBSON, V. (1997). SDP: Session Description Protocol. Rapport technique, IETF. draft-ietf-mmusic-sdp-05.
- [46] HERZOG, S., SHENKER, S., et ESTRIN, D. (1995). Sharing the Cost of Multicast Trees: An Axiomatic Analysis. *Proc. Communication Architectures, Protocols and Applications*.
- [47] HOLLAND, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- [48] HWANG, F. K., RICHARDS, D. S., et WINTER, P. (1992). The Steiner Tree Problem. *Annals of Discrete Mathematics*, vol. 53, pages 151–176.
- [49] HWANG, R. H. (1995). Adaptive Multicast Routing in Single Rate Loss Networks. *Proc. IEEE INFOCOM'95*, pages 571–578.
- [50] ITU-T (1993a). Text of Draft Recommendation I.362: B-ISDN ATM Adaptation Layer (AAL) Functional Description. Rapport technique, ITU.
- [51] ITU-T (1993b). Text of Draft Recommendation I.363: B-ISDN ATM Adaptation Layer (AAL) Specification. Rapport technique, ITU.
- [52] ITU-T (1995a). Text of Draft Recommendation I.326: Functional Architecture of Transport Networks Based on ATM. Rapport technique, ITU.

- [53] ITU-T (1995b). Text of Draft Recommendation I.371: Traffic Control and Congestion Control in B-ISDN. Rapport technique, ITU.
- [54] JAIN, R. (1997). Multipoint Communication over IP and ATM.
http://www.cis.ohio-state.edu/jain/cis788-97/h_ampt.htm.
- [55] KADIRIRE, J. (1994). Minimizing Packet Copies in Multicasting by exploiting Geographic Spread. *Computer Communications Review*, vol. 24, no 3.
- [56] KAUFMAN, J. S. (1981). Blocking in a Shared Resource Environment. *IEEE TRANSACTIONS ON COMMUNICATIONS*, vol. COM-29, no 10, pages 1474–1481.
- [57] KELLY, F. P. (1986). Blocking probabilities in large circuit-switched networks. *Adv. Appl. Prob.*, vol. 18, pages 473–505.
- [58] KELLY, F. P. (1988). Routing in Circuit-Switched Networks: Optimization, Shadow Prices and Decentralization. *Adv. Appl. Probl.*, vol. 20, pages 112–144.
- [59] KELLY, F. P. (1991). Effective Bandwidths at Multi-class Queues. *Queuing Systems*, vol. 9, pages 5–16.
- [60] KOMPELLA, V. P., PASQUALE, J. C., et POLYZOS, G. C. (1993). Multicast Routing for Multimedia Communication. *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 1, no 3, pages 286–292.
- [61] KOU, L., MARKOWSKY, G., et MERMAN, L. (1981). A Fast Algorithm for Steiner Trees. *Acta Informatica*, vol. 15, pages 141–145.

- [62] KUMAR, K. B. et JAFFE, M. J. (1983). Routing to Multiple Destinations in Computer Networks. *IEEE TRANSACTIONS ON COMMUNICATIONS*, vol. COM-31, no 3, pages 343–351.
- [63] KUMAR, S. et AGRAWAL, D. P. (1996). On Multicast for Shared-Memory Based ATM Switch Architecture. *IEEE NETWORK*, vol. 10, no 1, pages 34–39.
- [64] LABOURDETTE, J. F. P. et HART, G. W. (1992). Blocking Probabilities in Multitrafic Loss Systems: Insensitivity, Asymptotic Behavior and Approximations. *IEEE TRANSACTIONS ON COMMUNICATIONS*, vol. 40, no 8, pages 1355–1366.
- [65] LEE, T. et LIU, S. (1992). A Fair High Speed Copy Network for Multicast Packet Switch. Proc. *IEEE INFOCOM'92*, pages 886–894.
- [66] LIKHANOV, N., MAZUMDAR, R., et THÉBERGE, F. (1996). Calculation of Cell Loss Probabilities in Large Unbuffered Multiservice Systems. Proc. *IEEE ICC'97*, pages 560–564.
- [67] LUENBERGER, D. (1984). *Linear and non linear programming*. Addison-Wesley.
- [68] MAH, B. (1994). Measurements and Observations of IP multicast traffic. Rapport technique, Université de Californie, Berkeley. UCB/CSD-94-858.
- [69] MAKKI, K., PISSINOU, N., et FRIEDER, O. (1995). On Multicasting in a Communication Network. Proc. *IEEE INFOCOM'95*, pages 647–654.

- [70] MEDDEB, A., GIRARD, A., et ROSENBERG, C. (1997). The Impact of Tree Selection on the Design of Networks with Multipoint Connections. *Proc. IEEE GLOBECOM'97*, pages 1891–1897.
- [71] MEDDEB, A., GIRARD, A., et ROSENBERG, C. (1998a). Tree Selection Algorithm for Multirate Network Design with Point-to-Multipoint Communications. *soumis à IEEE/ACM TRANSACTIONS ON NETWORKING*.
- [72] MEDDEB, A., GIRARD, A., et ROSENBERG, C. (1998b). Tree Selection Algorithms for Point-to-Multipoint Call Routing in Multirate Networks. *soumis à IEEE/ACM TRANSACTIONS ON NETWORKING*.
- [73] MEDDEB, A., GIRARD, A., et ROSENBERG, C. (1999a). Tree Selection Heuristic for Network Design with Point-to-Multipoint Communications. *soumis à ITC-16*.
- [74] MEDDEB, A., GIRARD, A., et ROSENBERG, C. (1999b). Tree Selection Heuristic for Point-to-Multipoint Call Routing in Telecommunications Networks. *soumis à ITC-16*.
- [75] MINOUX, M. (1989). *Programmation mathématique, théorie et algorithmes*. Dunod.
- [76] MITRA, D., MORRISON, J. A., et RAMAKRISHNAN, K. G. (1996). ATM Network Design and Optimization: A Multirate Loss Network Framework. *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 4, no 4, pages 531–542.
- [77] MOGHÉ, P. et RUBIN, I. (1997). Reserving for Future Clients in a Multipoint Application— Why and How. *IEEE JSAC*, vol. 15, no 3.

- [78] MORGAN, S. (1998). The Internet and Local Telephone Network: Conflicts and Opportunities. *IEEE COMMUNICATIONS MAGAZINE*, vol. 36, no 1, pages 42–48.
- [79] MOY, J. (1994). Multicast Routing Extensions for OSPF. *COMMUNICATIONS OF THE ACM*, vol. 37, no 8, pages 61–66.
- [80] MYERSON, R. B. (1991). *Game Theory*. Harvard University Press.
- [81] NONNENMACHER, J., BIRSACK W., E., et TOWSLEY, D. (1998). Parity-Based Loss Recovery for Reliable Multicast Transmission. *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 6, no 4, pages 349–361.
- [82] PARSA, M., ZHU, Q., et GARCIA, L. A. J. J. (1998). An Iterative Algorithm for Delay-Constrained Minimum Cost Multicasting. *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 6, no 4, pages 461–474.
- [83] PEJHAN, S., ELEFThERIADIS, A., et ANASTASSIOU, D. (1995). Distributed Multicast Address Management in the Global Internet. *IEEE JSAC*, vol. 13, no 8, pages 1445–1456.
- [84] PLESNIK, J. (1981). Heuristics for the Steiner Tree Problem in Graphs. *Mathematica Slovaca*, vol. 31, pages 155–163.
- [85] RAMANATHAN, S. (1996). Multicast Tree Generation in Networks with Asymmetric Links. *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 4, no 4, pages 337–344.

- [86] RAVINDRAN, K. et GONG, T. J. (1998). Cost Analysis of Multicast Transport Architectures in Multiservice Networks. *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 6, no 1, pages 94–109.
- [87] RÉGNIER, J., BLONDEAU, P., et CAMERON, W. H. (1983). Grade of Service of a Dynamic Call-Routing System. *Proc. 10th International Teletraffic Congress*.
- [88] ROUSKAS, G. N. et BALDINE, I. (1997). Multicast routing with end-to-end delay and delay variation constraints. *IEEE JSAC*, vol. 15, pages 332–345.
- [89] SCHOEN, U., HAMANN, J., JUGEL, A., KURZAWA, H., et SHMIDT, C. (1998). Convergence between Public Switching and the Internet. *IEEE COMMUNICATIONS MAGAZINE*, vol. 36, no 1, pages 50–65.
- [90] SHACHAM, N. et YOKOTA, H. (1997). Admission Control Algorithms for Multicast Sessions with Multiple Streams. *IEEE JSAC*, vol. 15, no 3, pages 557–566.
- [91] STALLINGS, W. (1996). IPv6: The new Internet Protocol. *IEEE COMMUNICATIONS MAGAZINE*, vol. 34, no 7, pages 96–108.
- [92] SUBRAMANIAN, S. et SOMANI, A. K. (1995). Multicasting in ATM Networks Using MINs. *Proc. ICCCN'95*, pages 142–149.
- [93] TAKAHASHI, H. et MATSUYAMA, A. (1990). An Approximate Solution for the Steiner Problem in Graphs. *Mathematica Japonica*, vol. 6, pages 573–577.
- [94] TALPADE, R. R., AMITRAGE, G. J., et AMMAR, M. H. (1996). Experience with Architectures for Supporting IP Multicast over ATM. <http://www.lirmm.fr/atm/articles.html>.

- [95] THOMSON, K., MILLER, G. J., et WILDER, R. (1997). Wide Area Internet Traffic Patterns and Characteristics. *IEEE NETWORK*, vol. 11, no 6, pages 10–23.
- [96] VERMA, D. C. et GOPAL, P. M. (1993). Routing Reserved Bandwidth Multipoint Connections. Proc. *Communication Architectures, Protocols and Applications*, pages 96–105.
- [97] WAITZMAN, D., PARTRIDGE, C., et DEERING, S. (1988). Distance Vector Multicast Routing Protocol. Rapport technique, IETF. RFC 1075.
- [98] WAXMAN, B. M. (1988). Routing of Multipoint Connections. *IEEE JSAC*, vol. 6, no 9, pages 1617–1622.
- [99] WHITE, P. P. (1997). RSVP and Integrated Services in the Internet: A Tutorial. *IEEE COMMUNICATIONS MAGAZINE*, vol. 35, no 5, pages 100–107.
- [100] WHITE, P. P. (1998). ATM Switching and IP Routing Integration: The Next Stage in Internet Evolution? *IEEE COMMUNICATIONS MAGAZINE*, vol. 36, no 4, pages 79–83.
- [101] WINTER, P. (1987). Steiner problem in networks. *Networks*, vol. 17, pages 129–167.

ANNEXE A

Preuve de la proposition de concentration de trafic

Rappelons d'abord la proposition:

Proposition A.0.1 *Nous supposons que la fonction f donnant la capacité d'un arc s à blocage constant sur cet arc est concave et strictement croissante. Alors, la capacité marginale ΔN_s , requise pour augmenter le trafic offert à l'arc s d'une quantité $\Delta a_s > 0$ à blocage constant sur cet arc, diminue à mesure que la capacité de l'arc augmente.*

Démonstration: Soit la fonction g définie sur \mathbb{R}^+ par

$$g(x) = f(x + \Delta a_s) - f(x).$$

La capacité marginale requise pour augmenter le trafic offert à l'arc s d'une quantité Δa_s à blocage constant sur cet arc est

$$\Delta N_s = g(a_s).$$

où a_s est le trafic offert à l'arc.

Dans la mesure où f est croissante, montrer que ΔN_s diminue à mesure que la capacité de l'arc N_s augmente est équivalent à montrer que ΔN_s diminue à mesure que a_s augmente. Il suffit alors de montrer que $g(x)$ est une fonction décroissante. Or, f est concave donc, $f''(x) < 0$ ce qui veut dire que

$$f'(y) < f'(x) \quad \forall y > x.$$

Donc, on a

$$g'(x) = f'(x + \Delta a_s) - f'(x) < 0$$

ce qui prouve que $g(x)$ est décroissante.

ANNEXE B

Preuve de la proposition du chemin unique

.

B.1 Résultats préliminaires

Lemme B.1.1 *Soit f une fonction continue et concave sur l'ensemble des réels \mathbb{R} satisfaisant $f(0) = 0$. Pour $0 \leq \alpha \leq 1$, et pour $a \in \mathbb{R}$, on a*

$$f(a) \leq f(\alpha a) + f((1 - \alpha)a). \quad (\text{B.1})$$

Démonstration: Puisque f est concave, on a

$$f(\alpha a + (1 - \alpha)b) \geq \alpha f(a) + (1 - \alpha)f(b) \quad \forall (a, b) \in \mathbb{R} \times \mathbb{R}. \quad (\text{B.2})$$

Posons $b = 0$. En utilisant le fait que $f(0) = 0$, on a

$$f(\alpha a) \geq \alpha f(a). \quad (\text{B.3})$$

En substituant $1 - \alpha$ à α , on obtient

$$f((1 - \alpha)a) \geq (1 - \alpha)f(a). \quad (\text{B.4})$$

En additionnant les deux termes de gauche et les deux termes de droite des équations (B.3) et (B.4), on obtient l'équation (B.1).

Lemme B.1.2 *Soit f une fonction continue et concave sur l'ensemble des réels \mathbb{R} satisfaisant $f(0) = 0$. Pour $(x, y) \in \mathbb{R}^+ \times \mathbb{R}^+$ tels que $y \leq x$, on a*

$$yf(zx) \leq xf(zy) \quad \forall z \in \mathbb{R}. \quad (\text{B.5})$$

Démonstration: On a égalité pour $x = 0$. Si $x > 0$, on veut montrer que

$$yf(zx) \leq xf(zy) \Leftrightarrow \frac{y}{x}f(zx) \leq f\left(\frac{y}{x}zx\right)$$

et la démonstration est complétée en utilisant l'équation (B.3) et en remarquant que $0 \leq y/x \leq 1$.

Lemme B.1.3 Soient n et m deux entiers satisfaisant $m \geq n \geq 1$. On a

$$n \leq m \left(\frac{m-1}{m} \right)^{m-n}. \quad (\text{B.6})$$

Démonstration: Définissons la suite $(u_m)_{m \geq n}$ de terme général

$$u_m = m \left(\frac{m-1}{m} \right)^{m-n}.$$

Propriété: La suite u_m est croissante.

Démonstration: On peut remarquer que

$$\frac{u_{m+1}}{u_m} = \begin{cases} 1 & \text{si } m = n = 1 \\ \left(\frac{m^2}{m^2-1} \right)^{m-n} \geq 1 & \text{sinon} \end{cases}$$

ce qui prouve que u_m est croissante. On peut alors écrire

$$u_n \leq u_m \quad \forall m \geq n.$$

En remarquant que $u_n = n$, on a ainsi montré l'inégalité (B.6).

B.2 Preuve de la proposition du chemin unique

Nous allons d'abord rappeler la proposition:

Proposition B.2.1 Nous dimensionnons un réseau de N nœuds ($N \geq 3$) à blocage constant et égal à B_0 sur tous les arcs du réseau (mono-débit). Nous supposons que la fonction f donnant la capacité d'un arcs s à blocage constant sur cet arc est concave et strictement croissante et que $f(0) = 0$. Nous voulons acheminer une quantité de

trafic a (trafic offert) d'un nœud A vers un nœud B donnés. Nous supposons que nous n'avons pas encore installé de capacité entre A et B et que nous allons utiliser des arcs qui n'écoulent pas de trafic actuellement. Dans ces conditions, il suffit que $B_0 \leq 1/(N - 1)$ pour qu'il soit plus économique d'écouler le trafic de A vers B sur un chemin unique, ayant le plus petit nombre possible d'arcs.

Démonstration:

1) **Cas de chemins ayant le même nombre d'arcs:** Nous supposons, sans perte de généralité, que nous avons le choix entre acheminer le trafic sur un chemin unique à n arcs ou sur deux chemins distincts à n arcs chacun.

Dans le cas où on utilise un seul chemin (chemin k_1), dans la mesure où nous n'avons pas encore installé de capacité entre A et B et que nous allons utiliser des arcs qui écoulent seulement le trafic de A vers B, selon notre modèle, le trafic écoulé par tous les arcs du chemin est le même. On peut alors écrire

$$\bar{a}_s = \bar{a}_0 \quad \forall s \in k_1 \quad (\text{B.7})$$

où

$$\bar{a}_0 = aQ^{k_1} \quad (\text{B.8})$$

et

$$Q^{k_1} = \prod_{s \in k_1} [1 - B_s] = (1 - B_0)^n \quad (\text{B.9})$$

est la probabilité de connexion sur le chemin k_1 (voir équations (3.2) et (3.4)). Ainsi, d'après l'équation (3.5), le trafic offert à tous les arcs est

$$a_s = a_0 \quad \forall s \in k_1. \quad (\text{B.10})$$

où

$$a_0 = a(1 - B_0)^{n-1}. \quad (\text{B.11})$$

Dans le cas où on utilise deux chemins (chemins k_1 et k_2) avec respectivement α et $1 - \alpha$ comme coefficients de partage de charge, le trafic écoulé par les arcs du chemin k_1 est

$$\bar{a}_s = \alpha \bar{a}_0 \quad \forall s \in k_1 \quad (\text{B.12})$$

et le trafic écoulé par les arcs du chemin k_2 est

$$\bar{a}_s = (1 - \alpha) \bar{a}_0 \quad \forall s \in k_2. \quad (\text{B.13})$$

On peut alors déduire que le trafic offert aux arcs du chemin k_1 est

$$a_s = \alpha a_0 \quad \forall s \in k_1 \quad (\text{B.14})$$

et que le trafic offert aux arcs du chemin k_2 est

$$a_s = (1 - \alpha) a_0 \quad \forall s \in k_2. \quad (\text{B.15})$$

Pour montrer qu'il est plus économique d'utiliser seulement un chemin à n arcs, il suffit de montrer que

$$nf(a_0) \leq nf(\alpha a_0) + nf((1 - \alpha) a_0). \quad (\text{B.16})$$

En effet, dans chaque chemin, on a n arcs dont la capacité est donnée par la fonction f définie dans l'équation (5.10) où $B_s = Ct. = B_0$. L'inégalité (B.16) peut être facilement déduite du lemme B.1.1.

2) Cas de chemins n'ayant pas le même nombre d'arcs: Nous allons supposer, sans perte de généralité, que nous avons le choix entre acheminer tout le trafic sur un seul chemin à n arcs ou sur deux chemins dont l'un contient n arcs (chemin k_1) et l'autre contient $m > n$ arcs (chemin k_2) avec respectivement α et $1 - \alpha$ comme coefficients de partage de charge.

Dans le cas où nous avons un chemin à m arcs (chemin k_2), il suffit de substituer m à n dans les équations (B.11) et (B.15) pour trouver le trafic offert aux arcs du chemin.

Pour montrer qu'il est plus économique d'utiliser seulement le chemin à n arcs, il suffit de montrer que

$$nf(a_0) \leq nf(\alpha a_0) + mf((1 - \alpha)(1 - B_0)^{m-n} a_0). \quad (\text{B.17})$$

On sait que

$$nf(a_0) \leq nf(\alpha a_0) + nf((1 - \alpha)a_0)$$

(voir équation B.16). Ainsi, pour montrer l'inégalité (B.17), il suffit de montrer que

$$\begin{aligned} nf(\alpha a_0) + nf((1 - \alpha)a_0) &\leq nf(\alpha a_0) + mf((1 - \alpha)(1 - B_0)^{m-n} a_0) \Leftrightarrow \\ nf((1 - \alpha)a_0) &\leq mf((1 - \alpha)(1 - B_0)^{m-n} a_0). \end{aligned}$$

Pour alléger la notation, posons

$$x = (1 - \alpha)a_0$$

et

$$y = (1 - \alpha)(1 - B_0)^{m-n} a_0.$$

Remarquons que $y \leq x$. Ainsi, nous voulons montrer que

$$nf(x) \leq mf(y). \quad (\text{B.18})$$

D'après le lemme B.1.3, on a

$$n \leq m \left(\frac{m-1}{m} \right)^{m-n} \Leftrightarrow \frac{1}{m} \leq 1 - \left(\frac{n}{m} \right)^{\frac{1}{m-n}}. \quad (\text{B.19})$$

Or

$$\frac{1}{N-1} \leq \frac{1}{m} \quad (\text{B.20})$$

car dans un chemin de m arcs, on ne peut pas avoir plus que $N-1$ arcs dans un réseau de N nœuds. Comme nous avons posé comme condition

$$B_0 \leq \frac{1}{N-1}, \quad (\text{B.21})$$

en appliquant les inégalités (B.19–B.21), on obtient

$$B_0 \leq \frac{1}{N-1} \leq \frac{1}{m} \leq 1 - \left(\frac{n}{m}\right)^{\frac{1}{m-n}} \Rightarrow \frac{n}{m} \leq (1 - B_0)^{m-n}. \quad (\text{B.22})$$

D'après le lemme B.1.2, pour $z = 1$ et $x > 0$, puisque $y \leq x$ on a

$$\frac{y}{x} \leq \frac{f(y)}{f(x)}$$

Or,

$$\frac{y}{x} = (1 - B_0)^{m-n}$$

d'où

$$\frac{f(y)}{f(x)} \geq (1 - B_0)^{m-n} \geq \frac{n}{m} \Rightarrow nf(x) \leq mf(y)$$

ce qui prouve l'inégalité (B.18).